

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky**

Realizace sigma-delta modulátoru pro řízení výkonových měničů

**Implementation of the Sigma-Delta Modulator for
Controlling the Power Converter**

2018

Bc. Petr Dvořák

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání diplomové práce

Student: **Bc. Petr Dvořák**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T003 Aplikovaná elektronika
Téma: Realizace sigma-delta modulátoru pro řízení výkonových měničů
Implementation of the Sigma-Delta Modulator for Controlling the Power Converter
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Proveďte teoretický rozbor topologií sigma-delta modulátorů pro řízení výkonových měničů
2. Navrhněte mikropočítačový řídicí systém se sigma-delta modulátorem pro řízení výkonových měničů.
3. Realizujte navržený řídicí systém, včetně základního programového vybavení.
4. Pomocí měření ověřte vlastnosti navrženého sigma-delta modulátoru.

Seznam doporučené odborné literatury:


Podle pokynů vedoucího závěrečné práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Martin Sobek, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 27. dubna 2018


.....
podpis studenta

Poděkování

Rád bych poděkoval Ing. Martinu Sobkovi, Ph.D. za odbornou pomoc a konzultace při vytváření této diplomové práce.

Také bych rád poděkoval své rodině a přítelkyni za neustálou podporu během mého studia.

Abstrakt

Cílem této diplomové práce bylo navrhnout a realizovat mikropočítačový řídicí systém se sigma-delta modulátorem pro řízení výkonových měničů. První část této práce je zaměřena na teoretické seznámení s problematikou sigma-delta modulátorů pro řízení výkonových měničů. Následuje rozbor jednotlivých bloků navrženého mikropočítačového řídicího systému a dále je popsáno základní programové vybavení pro řídicí systém. Poslední část práce se zabývá ověřením funkce řídicího systému se sigma-delta modulátorem.

Klíčová slova

Digitální signálový procesor, FPGA, mikropočítačový řídicí systém, programovatelné hradlové pole, programové vybavení, sigma-delta modulace, sigma-delta modulátor

Abstract

The aim of this diploma thesis was to design and realize a microcontroller control system with a sigma-delta modulator for controlling power converters. The first part of this thesis focuses on the theoretical issues of sigma-delta modulators for controlling power converters followed by an analysis of individual blocks of the control system designed. Subsequently, the basic software equipment for the control system is described and the last part of the thesis deals with verification of the function of the control system with a sigma-delta modulator.

Key words

Digital Signal Controller, FPGA, Microcontroller Control System, Field Programmable Gate Array, Software Equipment, Sigma-Delta Modulation, Sigma-Delta Modulator

Obsah

Seznam použitých symbolů.....	- 7 -
Seznam použitých zkratek.....	- 8 -
Úvod.....	- 10 -
1 Teoretický rozbor topologií sigma-delta modulátorů pro řízení výkonových měničů....	- 11 -
1.1 Sigma-delta modulace.....	- 11 -
1.2 Sigma-delta modulátor 1. řádu.....	- 12 -
1.3 Sigma-delta modulátor 2. řádu.....	- 16 -
1.4 Alternativní struktury modulátorů 2. řádu	- 17 -
1.4.1 Struktura Boser-Wooley.....	- 17 -
1.4.2 Struktura Silva-Steensgaard	- 18 -
1.4.3 Struktura error-feedback.....	- 19 -
2 Návrh mikropočítačového řídicího systému.....	- 20 -
2.1 Kit eZdsp™ F28335.....	- 20 -
2.2 FPGA	- 21 -
2.3 Konfigurační paměť	- 22 -
2.4 Rozhraní AS a JTAG	- 23 -
2.5 Analogové výstupy ze signálového procesoru	- 23 -
2.6 D/A převodník.....	- 24 -
2.7 Analogové vstupy signálového procesoru	- 24 -
2.8 Rozhraní pro připojení rotačního optického enkodéru.....	- 26 -
2.9 Digitální výstupy programovatelného hradlového pole	- 27 -
2.10 Univerzální vstupy	- 28 -
2.11 Oscilátor	- 28 -
2.12 Napájení mikropočítačového řídicího systému	- 29 -
2.13 Programátor.....	- 31 -
2.14 Realizace mikropočítačového řídicího systému	- 31 -
2.15 Popis I/O konektorů mikropočítačového řídicího systému	- 32 -
3 Základní programové vybavení.....	- 35 -
3.1 Programové vybavení signálového procesoru.....	- 35 -
3.1.1 Cmd soubor pro paměťovou zónu 6	- 35 -
3.1.2 Hlavičkový soubor	- 35 -

3.1.3	Zdrojové soubory	- 36 -
3.2	Programové vybavení FPGA	- 41 -
3.2.1	Komunikace FPGA se signálovým procesorem.....	- 41 -
3.2.2	Sigma-delta modulátor 1. řádu	- 44 -
3.2.3	Pin planner.....	- 45 -
4	Ověření funkce řídicího systému se sigma delta modulátorem.....	- 46 -
	Závěr	- 51 -
	Použitá literatura	- 52 -
	Seznam příloh.....	- 53 -

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
A	[V]	Amplituda vstupního signálu
f	[Hz]	Frekvence
f_{max}	[Hz]	Maximální hodnota frekvence
f_{vz}	[Hz]	Vzorkovací frekvence
I_{out}	[A]	Výstupní proud
MP	[%]	Modulační poměr
MP_{vyp}	[%]	Modulační poměr vypočtený
T_{vz}	[s]	Vzorkovací perioda
U_d	[V]	Napětí v meziobvodu
U_{out}	[V]	Výstupní napětí
$U_{out(vyp)}$	[V]	Výstupní napětí vypočtené
$U_{sin(ef)}$	[V]	Efektivní hodnota vstupního sinusového signálu
$U_{noise(ef)}$	[V]	Efektivní hodnota kvantizačního šumu
X	[%]	Modulační chyba
$\Sigma\text{-}A$		Sigma-delta
N		Počet bitů převodníku
N_E		Efektivní rozlišení modulátoru vyjádřeno počtem bitů
M		Řád modulátoru
Z		Stav vysoké impedance

Seznam použitých zkratek

Zkratka	Význam
A/D	Analog/Digital (analogově digitální)
AS	Active Serial
CLK	Clock Signal (hodinový signál)
CS	Chip Select
D/A	Digital/Analog (digitálně analogový)
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPIO	General-purpose input/output (univerzální vstupy/výstupy)
FPGA	Field Programmable Gate Array (programovatelné hradlové pole)
IDE	Integrated Development Environment (vývojové prostředí)
I/O	Input/Output (vstup/výstup)
JTAG	Join Test Action Group
LED	Light-Emitting Diode (světlo vyzařující dioda)
LEs	Logic Elements (logické elementy)
HDL	Hardware Description Language (programový jazyk popisující hardware)
NTF	Noise Transfer Function (přenosová funkce šumu)
OSR	Oversampling Ratio (poměr převzorkování)
RAM	Random Access Memory (paměť s libovolným přístupem)
RD	Read (čtení)
SCI	Serial Communication System (sériové komunikační rozhraní)
SNR	Signal-To-Noise Ratio (odstup signálu od šumu)
SRAM	Static Random Access Memory (statická paměť)
STF	Signal Transfer Function (přenosová funkce signálu)
USB	Universal Serial Bus (univerzální sériová sběrnice)
VHDL	Very High Speed Integrated Circuit Hardware Description Language (programový jazyk popisující hardware upravený pro velmi rychlé integrované obvody)
WE	Write Enable (povolení zápisu)

XINTF	External Interface (externí rozhraní)
--------------	---------------------------------------

Úvod

Tato diplomová práce se zabývá vývojem mikropočítačového řídicího systému se sigma-delta modulátorem pro řízení výkonových měničů. Pro řízení výkonových měničů je v dnešní době již nevýhodné používat složitá analogová zapojení, proto se dnes využívá moderní mikropočítačové techniky, která využívá velmi rychlé signálové procesory, programovatelná hradlová pole atd.

Pro mikropočítačový řídicí systém byl na doporučení vedoucího práce zvolen vývojový kit eZdspTM F28335 vybavený signálovým procesorem TMS320F28335 od firmy Texas Instruments a programovatelné hradlové pole EP2C5T144C8N z rodiny Cyclone II od firmy ALTERA.

První kapitola se zabývá rozбором topologií sigma-delta modulátorů pro řízení výkonových měničů. Je zde vysvětlena problematika sigma-delta modulace spolu se základními pojmy potřebnými k pochopení. Dále jsou postupně rozebrány vhodné typy sigma-delta modulátorů pro řízení výkonových měničů.

V druhé kapitole je podrobně popsán návrh mikropočítačového systému, který se skládá ze dvou desek plošných spojů. První deska je již zhotovený kit eZdspTM F28335 se signálovým procesorem a druhá rozšiřující deska s programovatelným hradlovým polem. Komunikace mezi signálovým procesorem a FPGA je zajištěna vnější sběrnici procesoru.

Ve třetí kapitole je popsáno základní programové vybavení jak pro signálový procesor, tak pro programovatelné hradlové pole. Programové vybavení se skládá z programů pro správnou komunikaci mezi zařízeními a sigma-delta modulátorem 1. řádu pro ověření funkce řídicího systému.

Poslední kapitola zahrnuje ověření funkce mikropočítačového řídicího systému se sigma-delta modulátorem. Ověření bylo provedeno měřením na snižujícím měniči napětí řízeným navrženým sigma-delta modulátorem.

1 Teoretický rozbor topologií sigma-delta modulátorů pro řízení výkonových měničů

Tato kapitola se zabývá sigma-delta modulátory pro řízení polovodičových měničů. Pro tyto aplikace jsou vhodné modulátory pouze s jednobitovým kvantizérem, jejichž výstup nabývá pouze dvou hodnot. Jednobitový kvantizér je vhodné používat pouze v modulátorech 1. a 2. řádu, protože při použití jednobitového kvantizéru v modulátorech vyšších řádů nastávají problémy se stabilitou.

1.1 Sigma-delta modulace

Sigma-delta (Σ - Δ) modulace vychází z principu delta modulace, která patří do skupiny impulzových kvantovaných modulací.

Modulace (Σ - Δ) se skládá ze tří základních kroků:

- Vzorkování
- Kvantování
- Kódování

Vzorkování je prováděno pomocí vzorkovače, který pracuje na principu metody Sample and Hold. Tento obvod nejprve ze vstupního analogového signálu sejme vzorek (Sample) a následně si zapamatuje jeho hodnotu (Hold) do dalšího vzorkovacího okamžiku. Jednotlivé vzorky jsou od sebe vzdáleny o dobu odpovídající vzorkovací periodě, která musí být zvolena tak, aby byla zaručena dostatečná hustota vzorků. Vzorkovací perioda musí být dodržena, aby nevznikaly ztráty informace a nedošlo ke zkreslení při převodu spojitého signálu na diskretní. Vzorkovací perioda se volí podle Shannon-Kotělnikova teorému, který říká, že minimální vzorkovací frekvence musí být při nejmenším dvojnásobná, než jaká je maximální frekvence obsažená ve vzorkovacím signálu. Při nedodržení této podmínky by nebylo možné vstupní signál zrekonstruovat. [2, 3]

$$T_{VZ} = \frac{1}{f_{VZ}} \quad (1.1)$$

kde:

T_{VZ} (s) je vzorkovací perioda,
 f_{VZ} (Hz) je vzorkovací frekvence.

$$f_{VZ} \geq 2 \cdot f_{max} \quad (1.2)$$

kde:

f_{max} (Hz) je maximální hodnota frekvence obsažená ve vstupním signálu.

V praxi se však doporučuje použít vzorkovací kmitočet třikrát až pětikrát větší než maximální kmitočet obsažený v signálu.

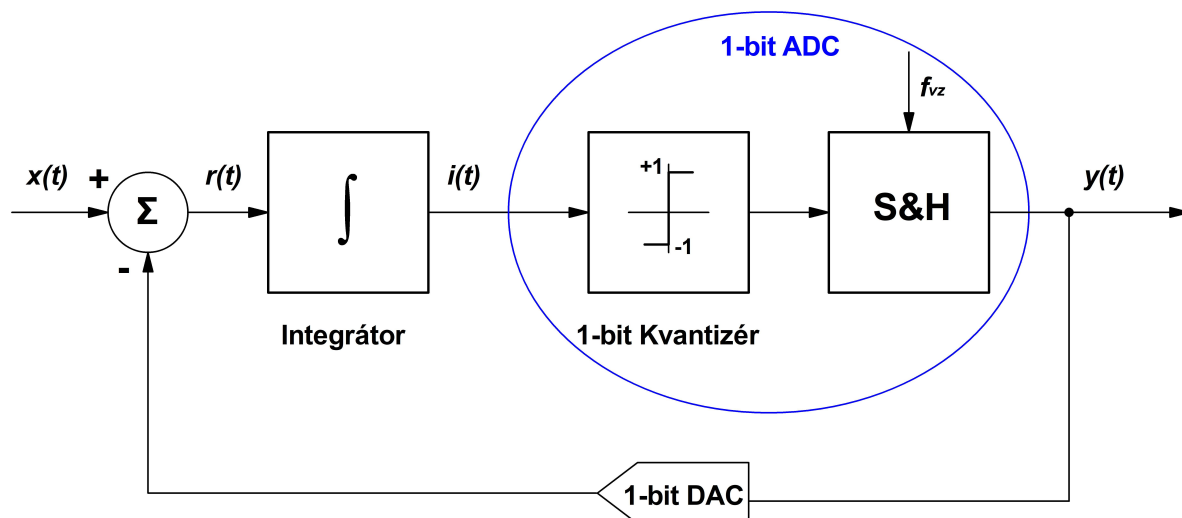
Kvantování je proces, kdy jednotlivým vzorkům impulzního signálu jsou přiřazeny hodnoty, které náležejí k nejbližší kvantizační úrovni.

Kódování je proces, který převádí kvantizační úrovně společně se vzorkovací frekvencí na zakódovanou informaci reprezentovanou binárním číslem. Kódovaný signál se rozděluje s ohledem na napěťový rozsah kvantovaného signálu. Může mít buď bipolární, nebo unipolární charakter. Unipolární kódování reprezentují pouze kladné hodnoty kódovaného signálu, naopak u bipolárního kódování jsou napěťové hodnoty kódovaného signálu záporné i kladné, kde záporné reprezentují nejnižší binární slovo.[4]

Sigma-delta (Σ - Δ) modulace nachází využití především v oblasti A/D převodníků, které se vyznačují velkým bitovým rozlišením, tedy i velkou přesností, ovšem na úkor rychlosti převodu.

1.2 Sigma-delta modulátor 1. řádu

Sigma-delta modulátor 1. řádu se využívá pro svou jednoduchou konstrukci. Je sestaven z rozdílového členu, integrátoru, jednobitového kvantizéru, obvodu Sample&Hold a jednobitového D/A převodníku, viz obr. 1.1.



Obrázek 1.1: Sigma-delta modulátor 1. řádu

Jednobitový kvantizér představuje komparátor s komparační úrovní v nule a společně s obvodem S&H, který je často reprezentován klopným obvodem D, tvoří jednobitový A/D převodník.

Funkce modulátoru vyplývá ze zapojení, viz obr. 1.1, kde výstupní signál $y(t)$ je převeden zpětnou vazbou přes D/A převodník do rozdílového členu zpět na vstup. Tady je signál $y(t)$ odečten od vstupního signálu $x(t)$ a jeho rozdíl $r(t)$ je přiveden do integrátoru. Výstup integrátoru $i(t)$ je přiveden do kvantizéru. Pokud bude výstup integrátoru $i(t)$ kladný, bude výstupní úroveň kvantizéru rovna +1. Naopak pokud bude výstup integrátoru záporný, bude výstupní hodnota kvantizéru -1. Výstup kvantizéru vstupuje do obvodu S&H, který vzorkuje signál vzorkovací frekvencí $OSR \cdot f_{vz}$ a pamatuje si jeho hodnotu do dalšího vzorkovacího okamžiku. Při vzorkování je důležité dodržet podmínku Shannon-Kotělnikova (Nyquistova) teorému, viz rovnice (1.2). Modulátor (Σ - Δ) však využívá tzv. převzorkování (z angl. oversampling), udávané koeficientem OSR (oversampling ratio). To znamená,

že jeho vzorkovací kmitočet je několikanásobně větší než u klasických převodníků. V praxi se koeficient OSR využívá o velikosti desítek až stovek.

$$OSR = \frac{f_{VZ}}{2 \cdot f_{max}} \quad (1.3)$$

Při kvantizačním procesu vzniká kvantizační chyba známá jako kvantizační šum. Tato chyba nastává, jelikož většina navzorkovaných hodnot neodpovídá kvantizačním úrovním. Kvantizační šum je nejčastěji vyjádřen jako rozdíl mezi kvantovaným signálem a signálem původním. Přesnost pro sinusový signál na vstupu převodníku se určuje odstupem signálu od šumu tzv. SNR [dB] (z angl. signal-to-noise ratio). Pro ideální A/D převodník s Nyquistovým vzorkováním je tento signál popsán vztahem 1.4. [1]

$$SNR[dB] = 20 \log \frac{U_{sin(ef)}}{U_{Noise(ef)}} = 6,02N + 1,76 \quad (1.4)$$

kde je:

$U_{sin(ef)}$	efektivní hodnota vstupního sinusového signálu,
$U_{Noise(ef)}$	efektivní hodnota kvantizačního šumu,
N	počet bitů převodníku.

Vztah pro SNR sigma-delta převodníku se upraví převzorkováním na [5]:

$$SNR[dB] = 20 \log \frac{U_{sin(ef)}}{U_{Noise(ef)}} = 6,02(N + N_E) + 1,76 \quad (1.5)$$

kde je:

N	počet bitů převodníku,
N_E	efektivní rozlišení modulátoru vyjádřeno počtem bitů.

Efektivní hodnota rozložení modulátoru N_E se vypočte dle vztahu [5]:

$$N_E = \frac{1}{6,02} \left[(20M + 10) \log OSR - 20 \log \left(\frac{\pi^M}{A\sqrt{2M+1}} \right) \right] \quad (1.6)$$

kde je:

M	řád modulátoru,
A	amplituda vstupního signálu.

Pro výpočet efektivní hodnoty N_E modulátoru 1. řádu dosazujeme za $M=1$ a $A=1$

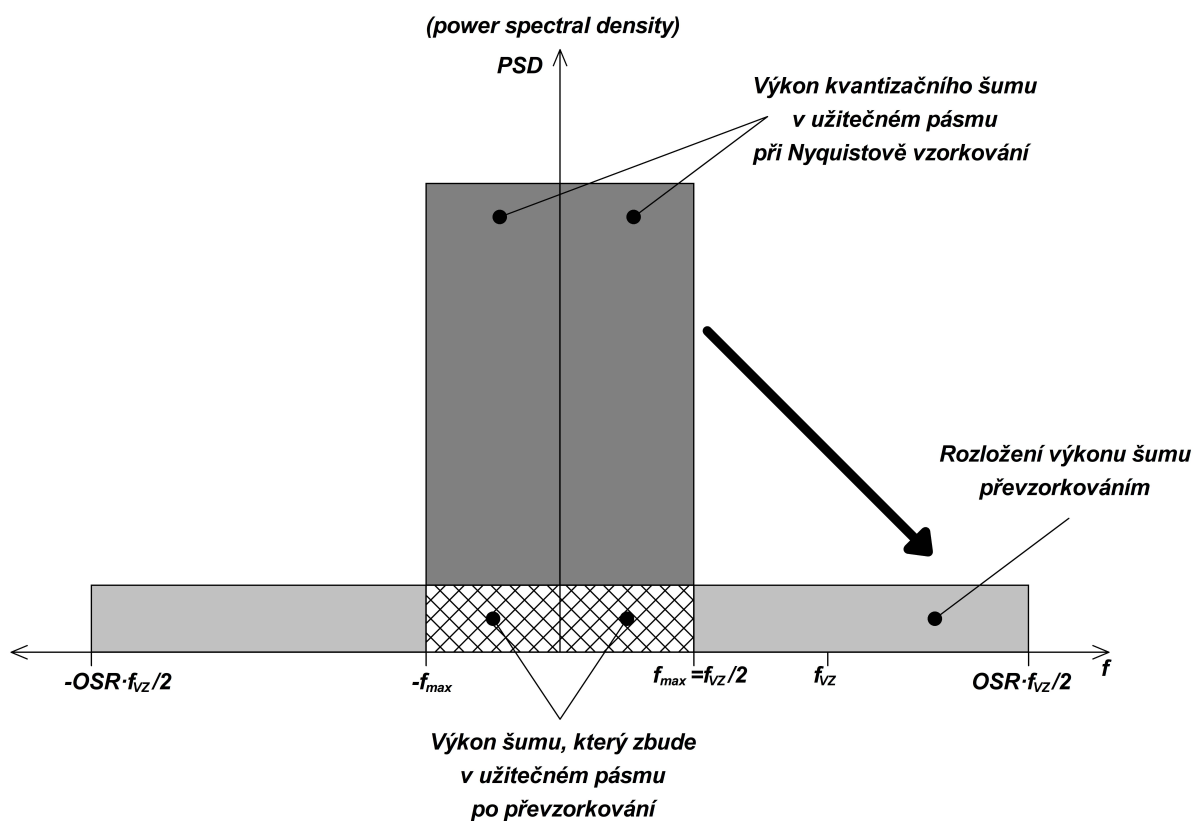
$$N_E = \frac{1}{6,02} \left[30 \log OSR - 20 \log \left(\frac{\pi}{\sqrt{3}} \right) \right] \quad (1.7)$$

Ze vztahu 1.7 je zřejmé, že při dosažení vyšší hodnoty koeficientu OSR a vyššího řádu modulátoru se N_E zvýší a s ním i celkové SNR. [5]

Modulátory sigma-delta disponují jednou unikátní vlastností, a to tzv. tvarováním kvantovacího šumu. V těchto modulátorech lze manipulovat se spektrálním rozložením kvantizačního šumu ve výstupním číslicovém signálu. To znamená, že jeho snahou je potlačení šumu v užitečném pásmu signálu. Tvarování kvantovacího šumu se provádí pomocí dvou metod, které působí zároveň, a jejich efekt se ve výsledku sečte [1]:

- rozložením výkonu šumu převzorkováním,
- tvarováním šumu pomocí vhodně zvolené přenosové funkce *STF* (z angl. Signal Transfer Function) v hlavní větvi převodníku (tzn. použití převodníku vyššího řádu).

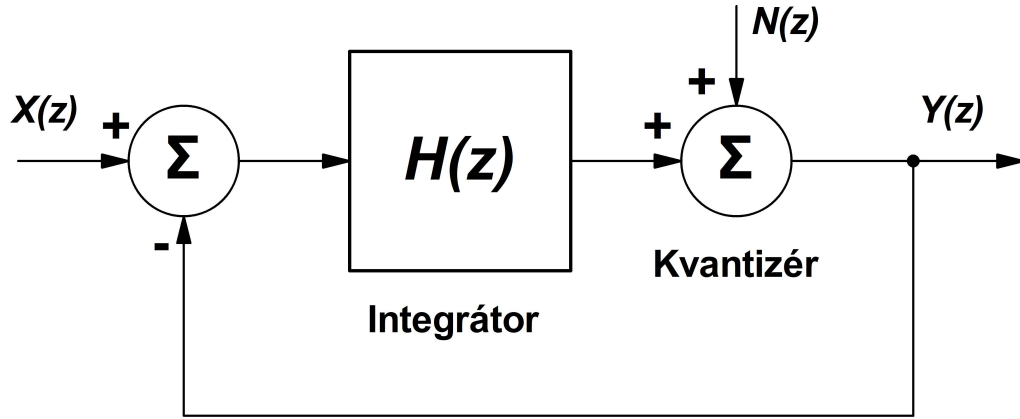
Oproti běžným převodníkům s Nyquistovým vzorkováním se vlivem převzorkování výkon kvantovacího šumu rozloží do širšího kmitočtového pásma, viz obr. 1.2.



Obrázek 1.2: Snížení výkonu kvantovacího šumu vlivem převzorkování

Z obrázku 1.2 vyplývá, že čím hustěji bude vzorkován vstupní signál, tím bude odstup šumu od užitečného signálu větší.

Pro určení přenosové funkce zavedeme lineární model Σ - Δ modulátoru v rovině z (diskrétním čase), viz obr. 1.3. V lineárním modelu je integrátor nahrazen blokem s obecným přenosem $H(z)$. Kvantizér je nahrazen součtovým členem, který přičítá ke kvantovanému signálu kvantovací šum $N(z)$. [1]



Obrázek 1.3: Blokové schéma lineárního modelu Σ - Δ modulátoru

Přenos lineárního modelu:

$$Y_{(z)} = H_{(z)} \cdot (X_{(z)} - Y_{(z)}) + N_{(z)} \quad (1.8)$$

Přenos integrátoru $H(z)$ je popsán rovnicí:

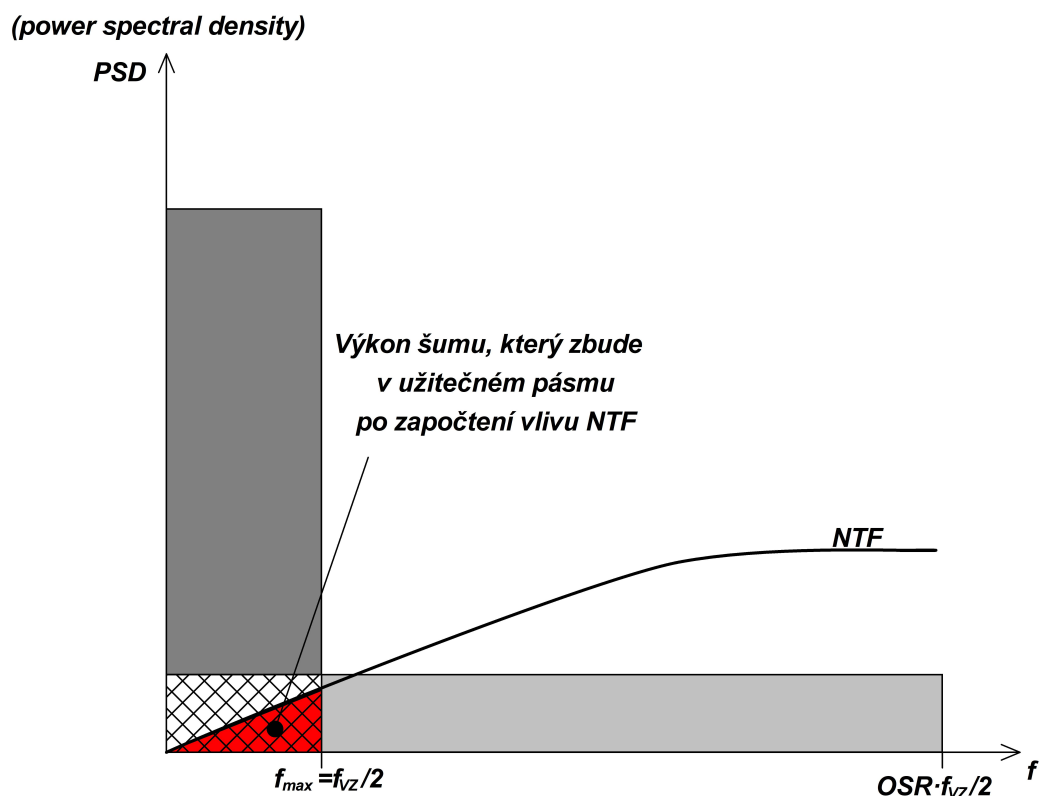
$$H_{(z)} = \frac{z^{-1}}{1 - z^{-1}} \quad (1.9)$$

Nyní lze určit přenosová funkce signálu STF a přenosová funkce šumu NTF (z ang. Noise Transfer Function).

$$STF_{(z)} = \frac{Y_{(z)}}{X_{(z)}} = \frac{H_{(z)}}{1 + H_{(z)}} = z^{-1} \quad (1.10)$$

$$NTF_{(z)} = \frac{Y_{(z)}}{N_{(z)}} = \frac{1}{1 + H_{(z)}} = 1 - z^{-1} \quad (1.11)$$

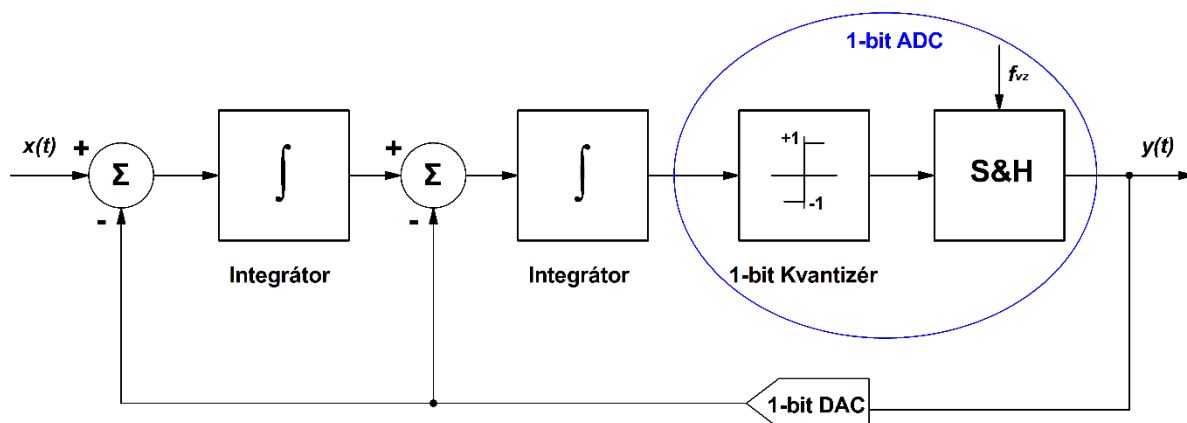
Ze vztahů je zřejmé, že funkce STF se chová jako zpožďovací člen a funkce NTF se chová jako filtr typu horní propust 1. řádu. Sigma-delta modulátor se vůči šumu sám chová jako filtr horní propust, který potlačuje šum v pásmu užitečného signálu. Na obr. 1.4 je zobrazeno tvarování kvantovacího šumu pomocí přenosové funkce NTF. [1]



Obrázek 1.4: Tvar přenosové funkce šumu NTF

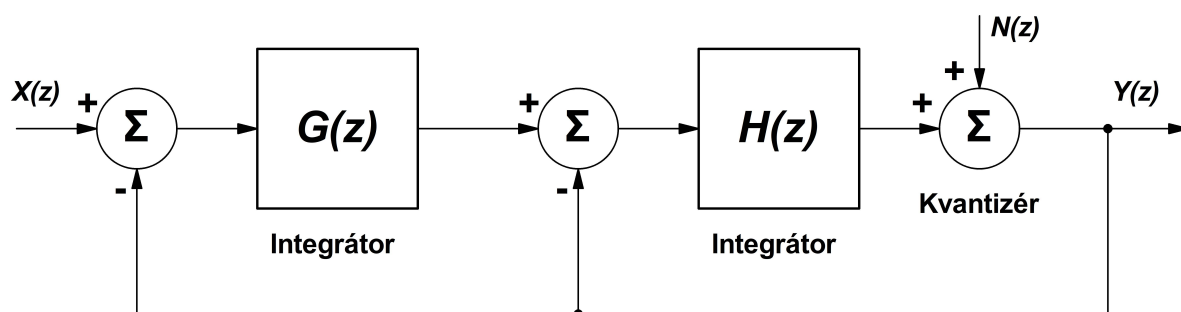
1.3 Sigma-delta modulátor 2. řádu

Sigma-delta modulátor 2. řádu je sestaven z modulátoru 1. řádu, který je doplněn o druhý integrátor a rozdílový člen, viz obr. 1.5.



Obrázek 1.5: Sigma-delta modulátor 2. řádu

Na obrázku 1.6 je sestaven lineární model modulátoru 2. řádu. Model se skládá ze dvou integrátorů, kde první integrátor je ideální (nemá zpoždění) a druhý je reálný (se zpožděním). [1]



Obrázek 1.6: Blokové schéma lineárního modelu modulátoru 2. řádu

Výpočet efektivní hodnoty rozlišení modulátoru 2. řádu dostaneme dosazením $M=2$ a $A=1$ do rovnice 1.6.

$$N_E = \frac{1}{6,02} \left[50 \log \text{OSR} - 20 \log \left(\frac{\pi^2}{\sqrt{5}} \right) \right] \quad (1.12)$$

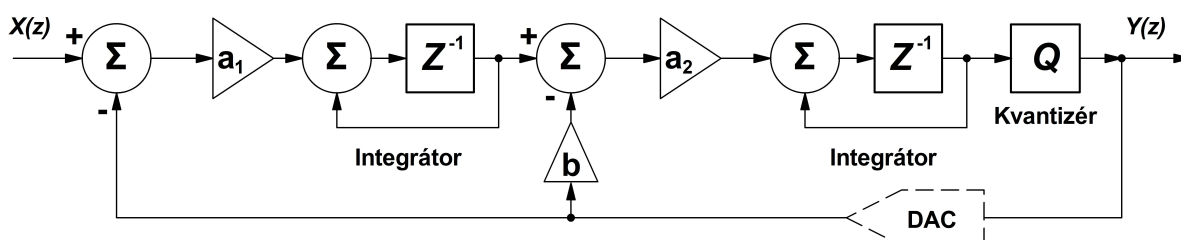
Ze vztahu 1.12 je zřejmé, že při dosažení vyšší hodnoty koeficientu OSR se N_E zvýší a sním i celkové SNR. Z toho vyplývá, že modulátor 2. řádu dosahuje stejných efektivních hodnot rozlišení, jako modulátor 1. řádu, již při nižších hodnotách OSR.[1]

1.4 Alternativní struktury modulátorů 2. řádu

Pro reálné aplikace se zavádějí alternativní struktury modulátoru 2. řádu, které obsahují integrátory se zpožděním. Tyto struktury mají velmi podobné vlastnosti jako modulátor 2. řádu a stejnou přenosovou funkci NTF . [3]

1.4.1 Struktura Boser-Wooley

Struktura obsahuje dva integrátory se zpožděním, které fungují nezávisle, čímž se snižují požadavky na jejich rychlost.[6]



Obrázek 1.7: Modulátor Boser-Wooley

Vztahy pro přenosové funkce STF a NTF [6]:

$$STF_{(z)} = \frac{a_1 a_2 z^{-2}}{D_{(z)}} \quad (1.13)$$

$$NTF_{(z)} = \frac{1 - z^{-1}}{D_{(z)}} \quad (1.14)$$

kde:

$$D_{(z)} = (1 - z^{-1})^2 + a_2 b_z^{-1} (1 - z^{-1}) + a_1 a_2 z^{-2} \quad (1.15)$$

Pro dosažení $STF_{(z)} = z^{-2}$ a $NTF_{(z)} = (1 - z^{-1})^2$ musí být splněny tyto podmínky:

$$\bullet \quad a_1 a_2 = 1 \quad (1.16)$$

$$\bullet \quad a_2 b = 2 \quad (1.17)$$

Jelikož máme tři parametry a jen dvě omezení, dá se to řešit mnoha způsoby, např.:

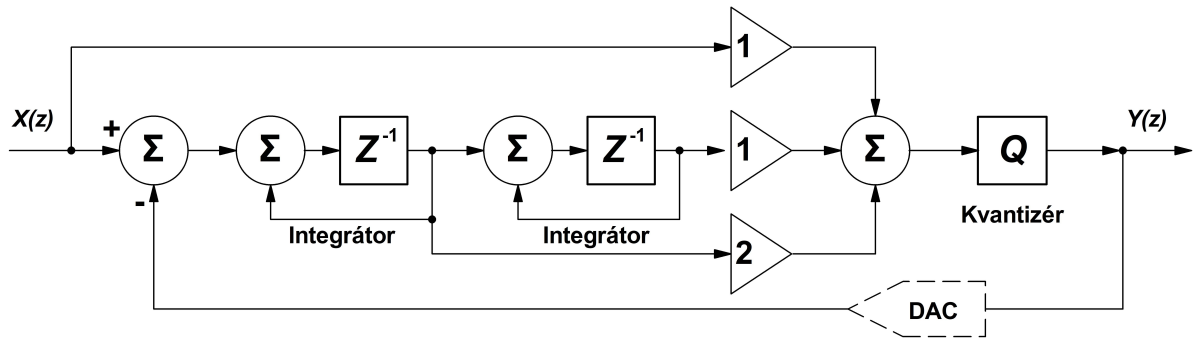
$$\bullet \quad a_1 = a_2 = 1 \text{ a } b = 2 \quad (1.18)$$

nebo

$$\bullet \quad a_1 = 0,5 ; a_2 = 2 ; b = 1 \quad (1.19)$$

1.4.2 Struktura Silva-Steensgaard

Tato struktura využívá dopředné vazby, kde vstupní signál je přiváděn do kvantizéru spolu s výstupy všech integrátorů. Digitální výstup je pak přiveden zpětnou vazbou zpět na vstup. Vstupní signál, vstupující do smyčky, obsahuje pouze tvarovaný kvantizační šum. Tím se snižují nároky na linearitu integrátorů, což je velká výhoda této struktury. [6]



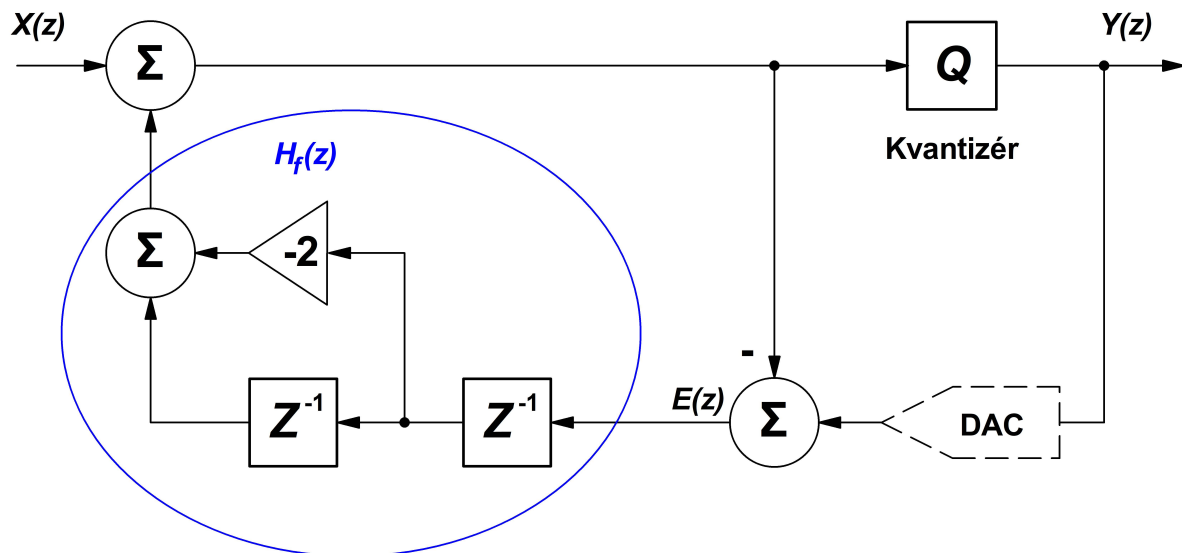
Obrázek 1.8: Modulátor Silva-Steensgaard

Přenos je dán rovnicí:

$$Y_{(z)} = X_{(z)} + (1 - z^{-1})^2 + E_{(z)} \quad (1.20)$$

1.4.3 Struktura error-feedback

Error-feedback má relativně jednoduchou strukturu, která je bohužel nepoužitelná pro analogové aplikace. Kvantizační chyba se získává rozdílem vstupního signálu interního A/D převodníku a výstupem D/A převodníku. Kvantizační chyba je následně přivedena zpětnovazební smyčkou přes filtr H_f zpět na vstup obvodu. Struktura error-feedback je dobře využitelná pro digitální účely. [6]



Obrázek 1.9: Struktura error-feedback

Přenos je dán rovnicí:

$$Y_{(z)} = E_{(z)} + X_{(z)} + H_{f(z)}E_{(z)} \quad (1.21)$$

Vztahy pro přenosové funkce STF a NTF :

$$STF_{(z)} = 1 \quad (1.22)$$

$$NTF_{(z)} = 1 + H_{f(z)} \quad (1.23)$$

Pro dosazení

$$NTF_{(z)} = (1 - z^{-1})^2 \quad (1.24)$$

platí:

$$H_{f(z)} = (1 - z^{-1})^2 - 1 = z^{-2} - 2z^{-1} \quad (1.25)$$

2 Návrh mikropočítačového řídicího systému

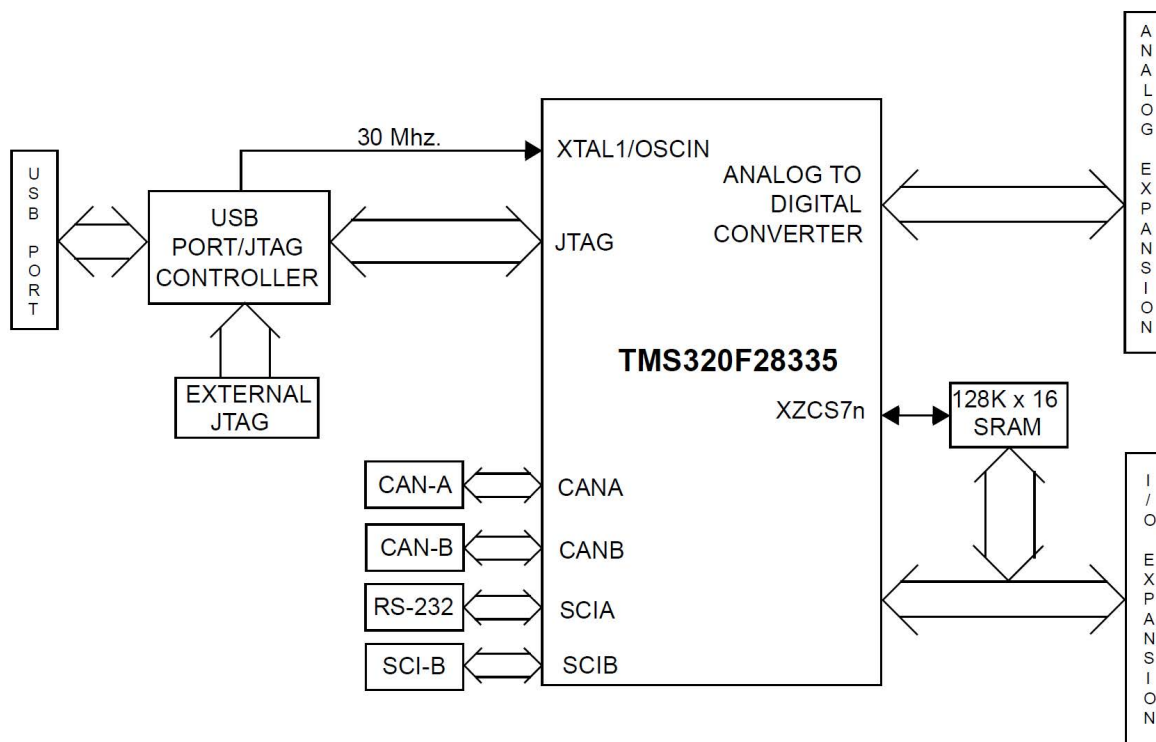
Na základě požadavků pro řízení polovodičových měničů byl vytvořen mikropočítačový řídicí systém, který obsahuje základní vybavení pro komunikaci, řízení a zpracování skutečných veličin. Systém je složen ze dvou částí. První část tvoří kit eZdspTM F28335 vybavený signálovým procesorem TMS320F28335 a druhá část je tvořena rozšiřující deskou s programovatelným hradlovým polem FPGA EP2C5T144C8N s dalšími základními obvody pro řízení polovodičových měničů. Komunikace mezi signálovým procesorem a FPGA je zajištěna vnější sběrnici procesoru. Vnější sběrnice se skládá z 16-ti datových vodičů, 10-ti adresových vodičů a 3 vodičů pro řídicí signály RD, WE a CS. Jednotlivé bloky navrženého systému budou popsány v následujících podkapitolách.

2.1 Kit eZdspTM F28335

Pro řídicí systém byl zvolen kit eZdspTM F28335 od firmy Texas Instruments, jehož blokové schéma je zobrazeno na obr. 2.1 a jeho základní informace o hardwaru jsou vypsány v tabulce 1.1. Kompletní dokumentace od výrobce je umístěna v příloze na CD.

Tabulka 1.1: Specifikace hardwarového vybavení kitu eZdspTM F28335 [7, 8]

Signálový procesor TMS320F28335	S 32bitovou plovoucí čárkou
Frekvence jádra	150 MHz
Integrovaná paměť RAM	68 KB
Integrovaná paměť FLASH	512 KB
Externí paměť SRAM	256 KB
Sběrnice	16bitová
Integrovaný A/D převodník	12bitový
Komunikační rozhraní	RS232, JTAG, SCI, CAN 2.0, mini USB
Rozšiřující konektory	Analog, I/O
Napájecí konektor	5V



Obrázek 2.1: Blokové schéma kitu eZdsp™ F28335 [7]

2.2 FPGA

Další část řídicího systému tvoří programovatelné hradlové pole EP2C5T144C8N z rodiny Cyclone II od firmy *ALTERA*, viz obr. 2.2.

Rodina Cyclone II navazuje na předchozí řadu obvodů Cyclone I. V této řadě byl navýšen počet logických elementů a vstupně-výstupních pinů. Tyto obvody jsou vyráběny 90nm technologií, která umožňuje vytváření složitých číslicových systémů a má velký vliv na snížení spotřeby a zvýšení výkonu.

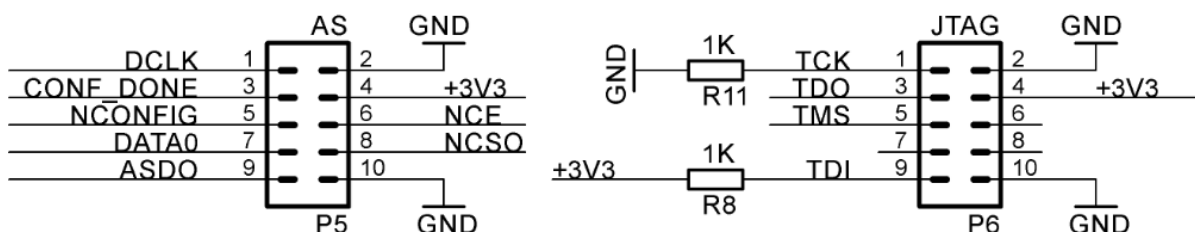
Základní vlastnosti obvodu EP2C5T144C8N:

Tabulka 1.2: Vlastnosti obvodu EP2C5T144C8N

Logické elementy (LEs)	4608
I/O piny	144
M4K RAM paměťové bloky	26
RAM celkově [bity]	199,808
Násobičky	13
Fázové závěsy	2

2.4 Rozhraní AS a JTAG

Na desce jsou k dispozici dvě programová rozhraní, s jejichž pomocí lze konfigurovat FPGA. Rozhraní AS slouží pro konfiguraci paměti, z které se nahrává program bezprostředně po zapojení napájení. Rozhraní JTAG slouží pro ladění a experimentování s programem.

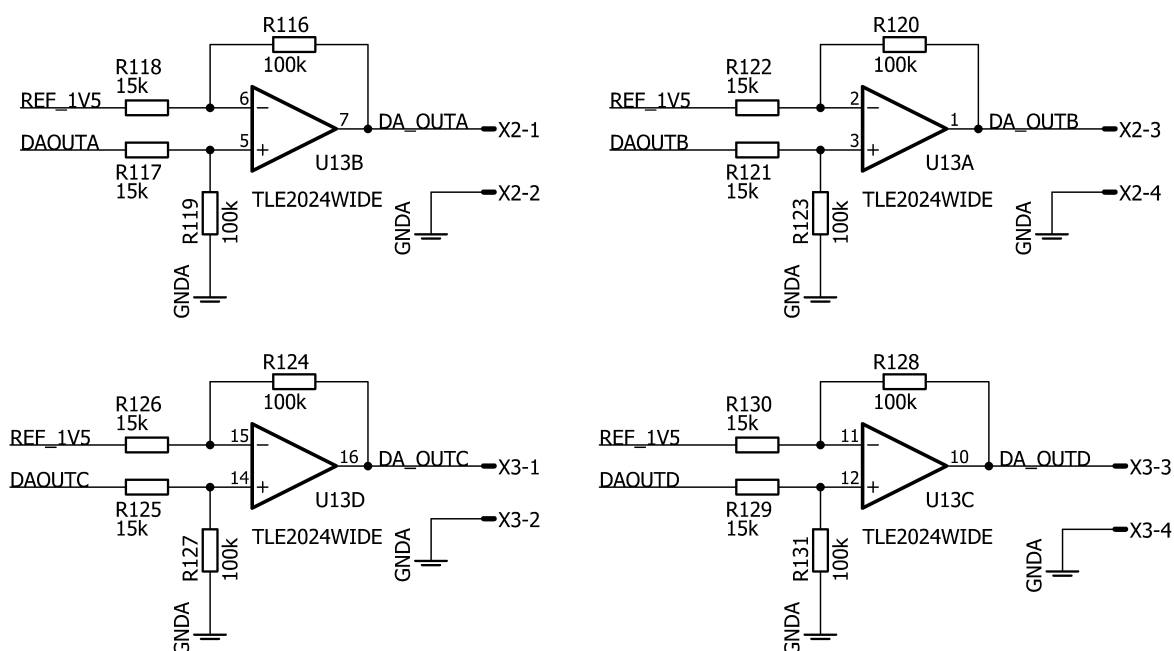


Obrázek 2.4: Programová rozhraní AS a JTAG

2.5 Analogové výstupy ze signálového procesoru

Analogové výstupy slouží k připojení externích měřících přístrojů, jako je např. osciloskop.

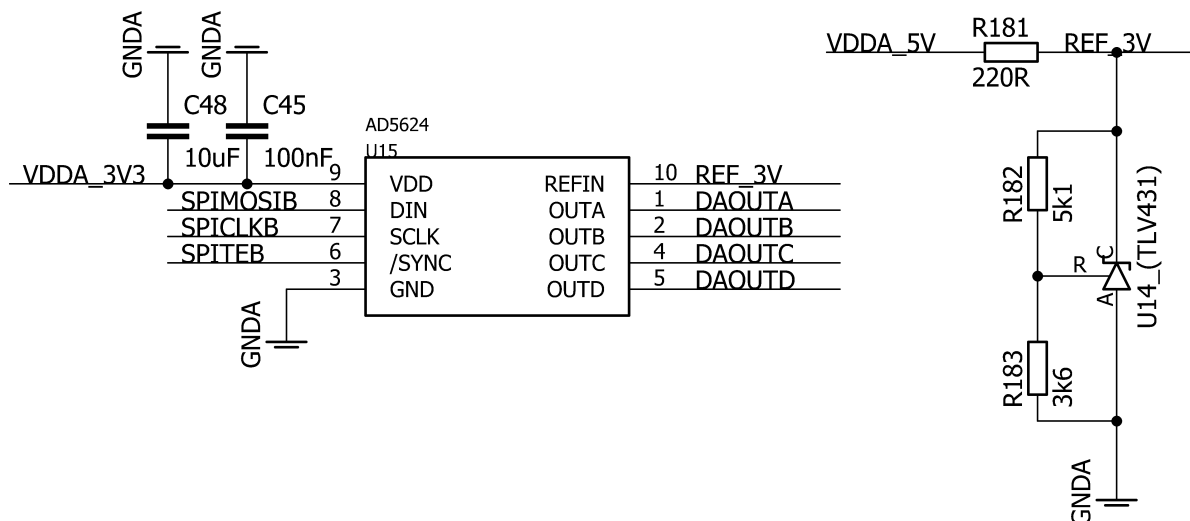
Analogové výstupy jsou vyvedeny ze signálového procesoru, přes sériové komunikační rozhraní SPI. Výstupní data jsou dále zpracována pomocí D/A převodníku AD5624, který disponuje čtyřmi analogovými výstupy. Samotné zapojení D/A převodníku bude popsáno v kapitole 2.6. Jelikož signálový procesor pracuje na napěťové úrovni +3,3V a výstupní napěťová úroveň D/A převodníku je nastavena na +3V, jsou výstupy vyvedeny na rozdílový zesilovač, který zajišťuje změnu napěťové úrovně z +3V na $\pm 10V$.



Obrázek 2.5: Analogové výstupy ze signálového procesoru

2.6 D/A převodník

Pro realizaci D/A převodníku byl zvolen převodník AD5624. Jedná se o 12bitový sériový D/A převodník, určený pro sériová rozhraní do 50MHz. Převodník má 4 výstupní kanály a vyznačuje se nízkou spotřebou 2,25mW. Pro nastavení výstupního rozsahu 0V až +3V je připojena napěťová reference +3V realizovaná obvodem TLV431.

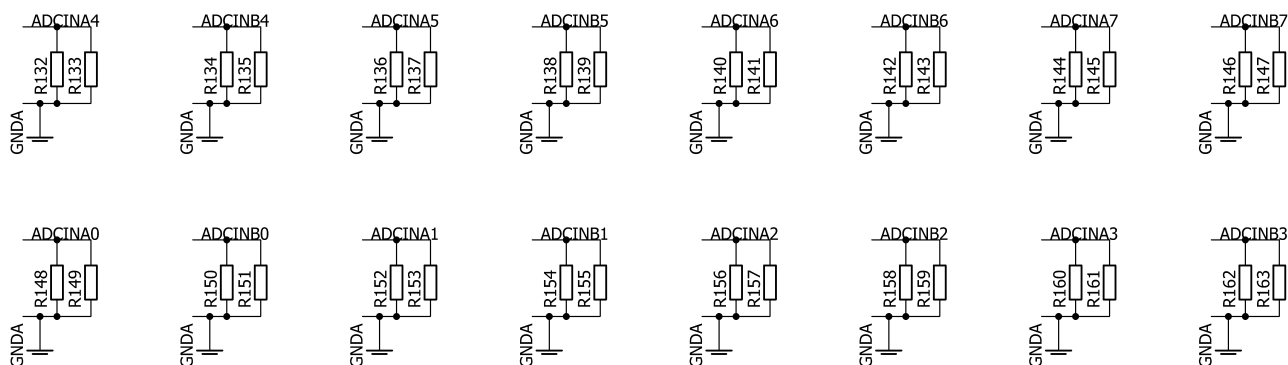


Obrázek 2.6: D/A převodník AD5624 s napěťovou referencí

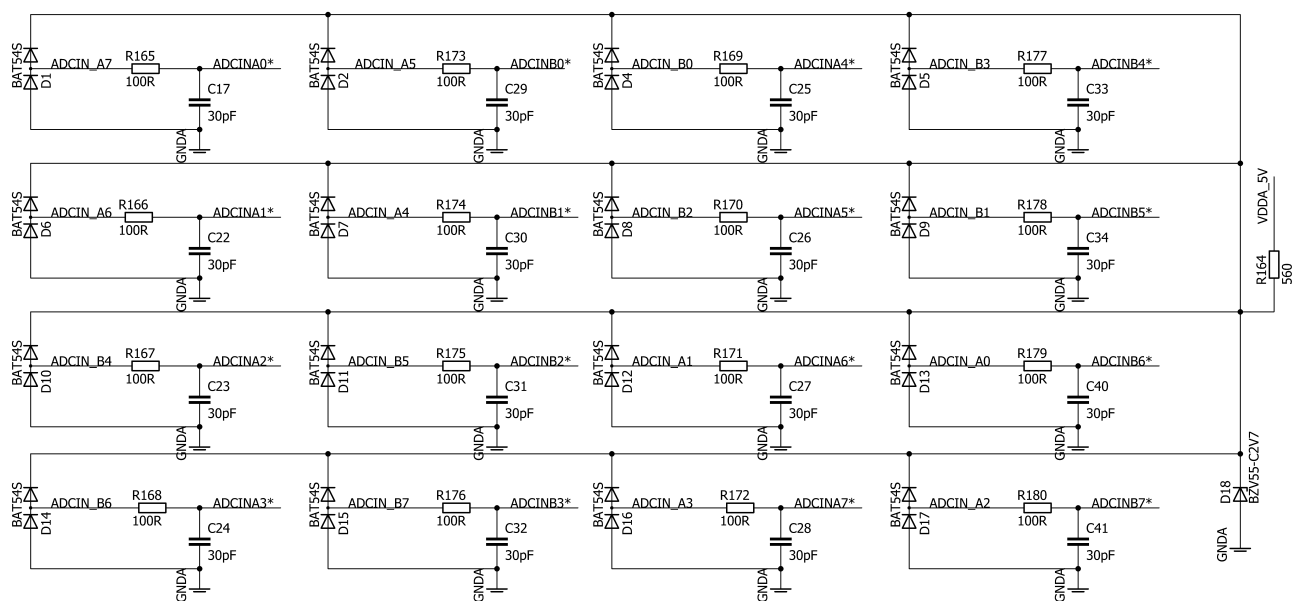
2.7 Analogové vstupy signálového procesoru

Analogové vstupy slouží k úpravě analogových signálů pomocí operační sítě na úroveň zpracovatelnou signálovým procesorem. Procesor následně převede analogový signál na digitální pomocí interního A/D převodníku pro další zpracování.

Pro připojení proudového čidla je možnost na vstupu osadit rezistor paralelně k zemi. Na desce je možnost osazení rezistorů v SMD nebo vývodovém provedení, viz obr. 2.7. Rezistor se volí podle rozsahu zvoleného čidla. Provedení SMD je pro menší výkon a vývodové provedení pro větší výkony. Změna úrovně vstupního analogového signálu je provedena rozdílovým zesilovačem s napěťovou referencí 1,5V. Na výstup z rozdílového zesilovače je připojen RC člen, tvořící filtr typu dolní propust. Rozsah výstupního napěťového signálu je omezen diodami pro ochranu vstupních analogových kanálů signálového procesoru, viz obr. 2.8.

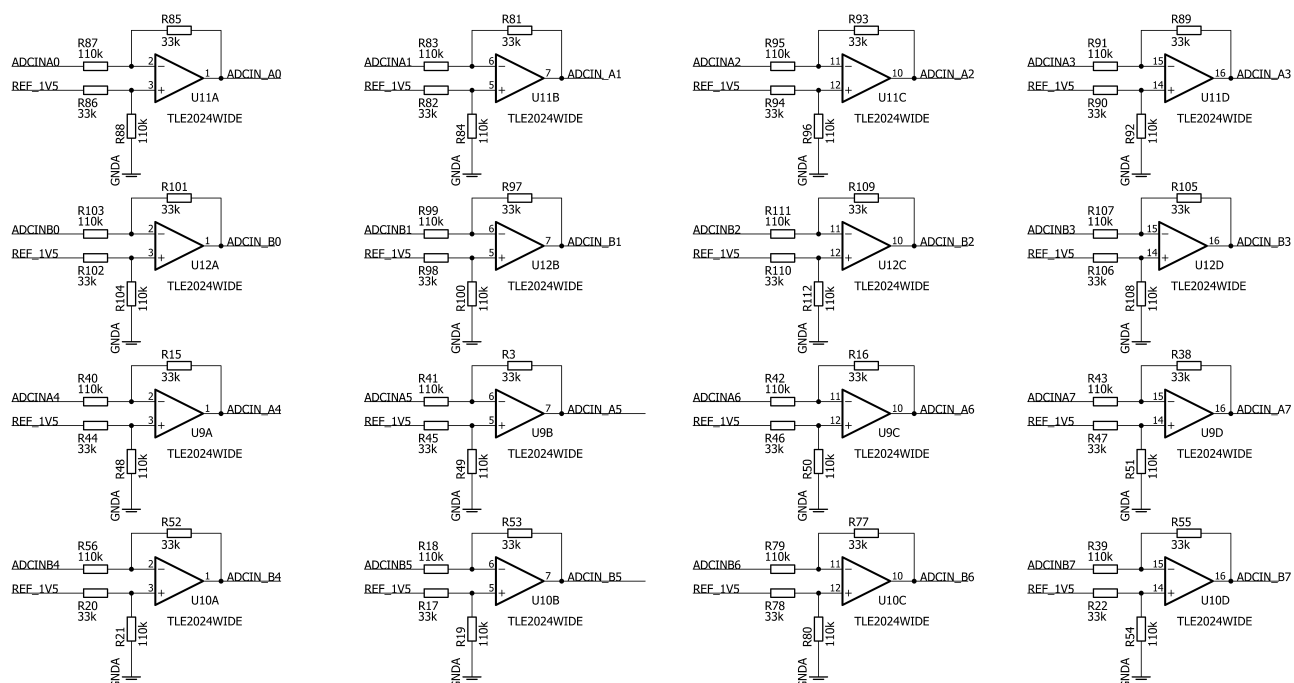


Obrázek 2.7: Měřicí rezistory

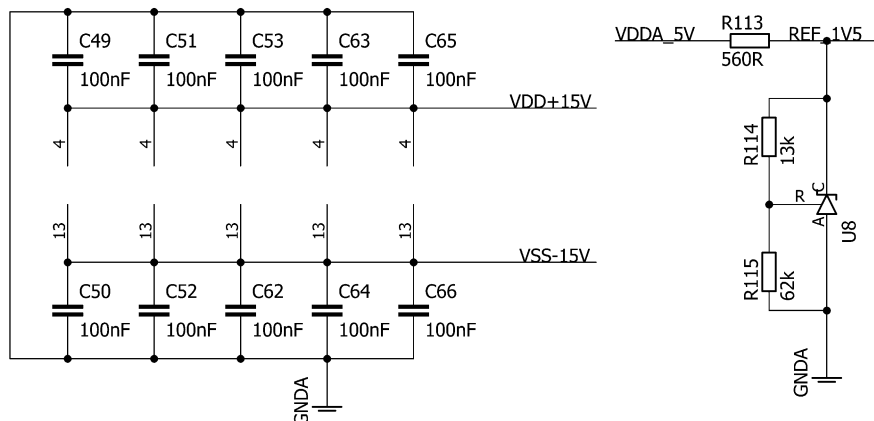


Obrázek 2.8: Filtr typu dolní propust s diodovou ochranou analogových vstupů TMS

Na obrázku 2.9 a 2.10 je schéma invertujícího rozdílového zesilovače s napěťovou referencí. Výstupní signál zesilovače je nastaven pro rozsah 0V až +3,3V. Po přivedení signálu na invertující vstup s amplitudou $U_{INV} = +5V$ je na výstupu zesilovače $U_{OUT} = 0V$ a při amplitudě $U_{INV} = -5V$ je na výstupu $U_{OUT} = +3,3V$.



Obrázek 2.9: Zapojení invertujícího rozdílového zesilovače



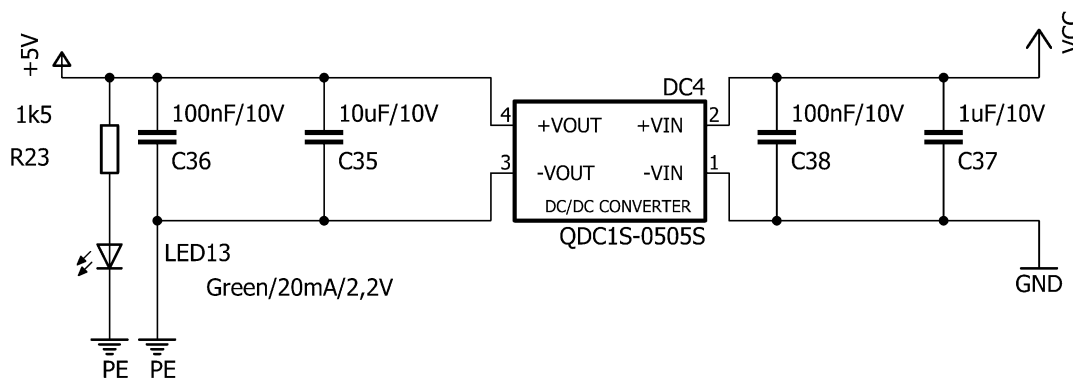
Obrázek 2.10: Filtrace napájení operačních zesilovačů s napětovou referencí 1,5V

2.8 Rozhraní pro připojení rotačního optického enkodéru

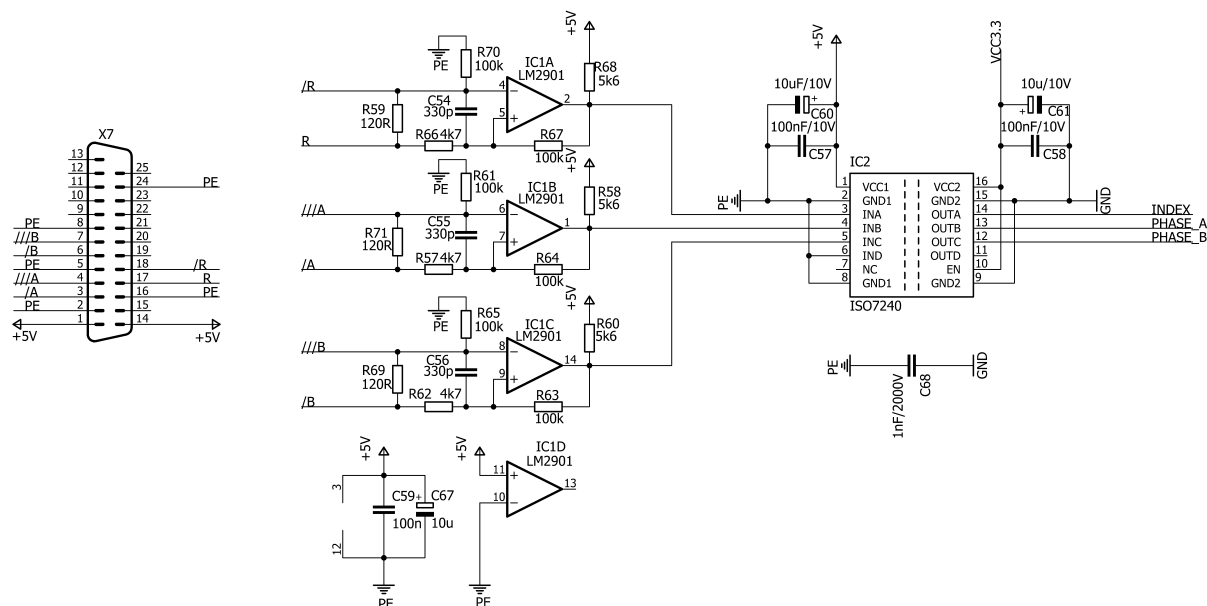
Jelikož signálový procesor TMS320F28335 disponuje perifériemi, které umožňují využití snímačů polohy a otáček, je na desku připojen enkodér.

Enkodér je zařízení, které je umístěno na hřídeli motoru a není tedy přímo na desce. Slouží k přesnému určení polohy a rychlosti otáčení motoru. Enkodér převádí rotační pohyb na digitální pulsní signál. Pro zajištění signálu bez rušení je nutné elektrický signál zesílit a elektricky zpracovat. Pro zlepšení kvality a stability signálu se využívá snímání v diferenciálním režimu, kdy se porovnávají dva téměř totožné signály s opačnou fází.[9]

Deska je vybavena 25pinovým konektorem CANON pro připojení snímače. Konektor je zapojen dle standardu. Signály z čidla jsou snímány diferenciálně z důvodu již zmíněnému potlačení šumu. Diferenciální signály jsou připojeny na vstupy operační sítě, kde se pomocí komparátorů vyhodnocují. Výstupní signály z komparátorů jsou připojeny k digitálnímu galvanickému izolátoru. Izolátor, zároveň s galvanickým oddělením, zajišťuje převod logické úrovně signálů z +5V na +3,3V. Výstupní signály jsou připojeny k signálovému procesoru. Toto zapojení je napájeno ze samostatného galvanicky odděleného zdroje tvořeného měničem ROE-0505S. Zdroj je vybaven signalizací v podobě zelené LED diody. Zapojení je doplněno o kondenzátor 1nF/2000V pro ochranu před elektrostatickým výbojem.



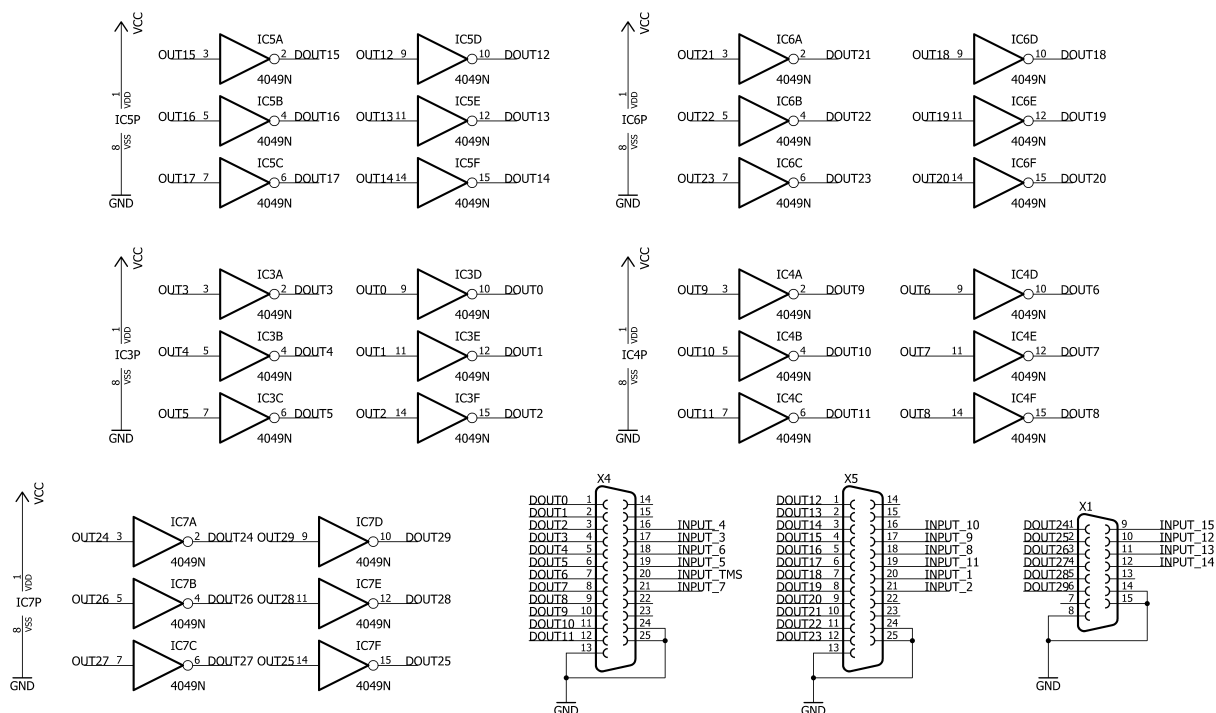
Obrázek 2.11: Galvanicky oddělený zdroj



Obrázek 2.12: Rozhraní pro připojení rotačního enkodéru

2.9 Digitální výstupy programovatelného hradlového pole

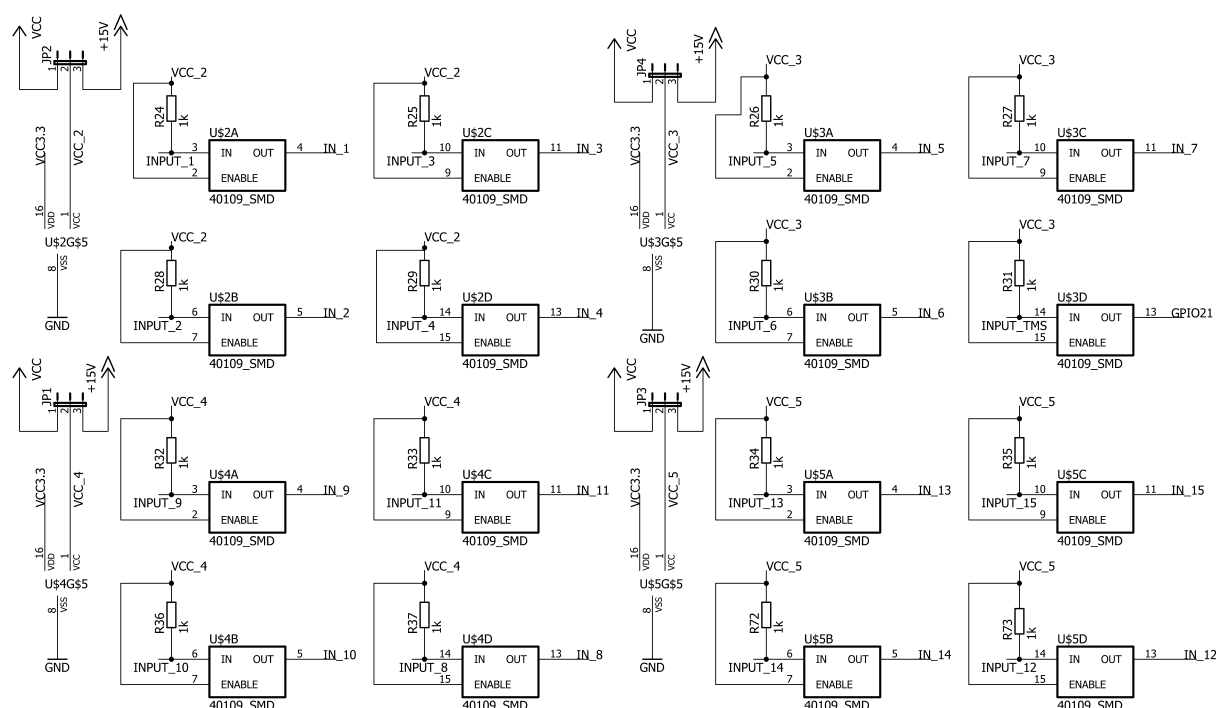
Z programovatelného hradlového pole je vyvedeno třicet výstupů, které mohou sloužit např. k připojení signálů řídicího systému k budičům měniče. Digitální výstupy jsou vybaveny obvody pro zesílení výkonu výstupního signálu. Deska je osazena precizními paticemi, do kterých je možnost zapojit dva typy hradel. Prvním typem je invertující hradlo CMOS 4049 a druhým typem neinvertující CMOS 4050. Tato hradla zesilují signál z +3,3V na +5V. Digitální výstupy jsou vyvedeny společně s univerzálními vstupy na konektory CANON, viz obr. 2.13.



Obrázek 2.13: Digitální výstupy programovatelného hradlového pole

2.10 Univerzální vstupy

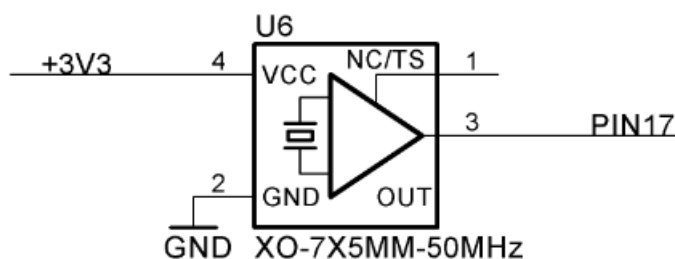
Univerzální vstupy slouží jako vstupy pro připojení vnějších obvodů. Na desce je připojeno patnáct univerzálních vstupů k programovatelnému hradlovému poli a jeden univerzální vstup připojený k signálovému procesoru. Ošetření vnějších signálů s větší napětíovou úrovní než +3,3V je provedeno obvodem CMOS 40109. Tento obvod pracuje jako level shifter, tedy převodník úrovní. Převodník je možné přepnout pro převod napětíové úrovně +15V nebo +5V na +3,3V. Přepnutí se provádí pomocí jumperů, viz obr. 2.14. Vstupy se dají použít např. pro zpětné hlášení poruchy z budičů.



Obrázek 2.14: Univerzální vstupy

2.11 Oscilátor

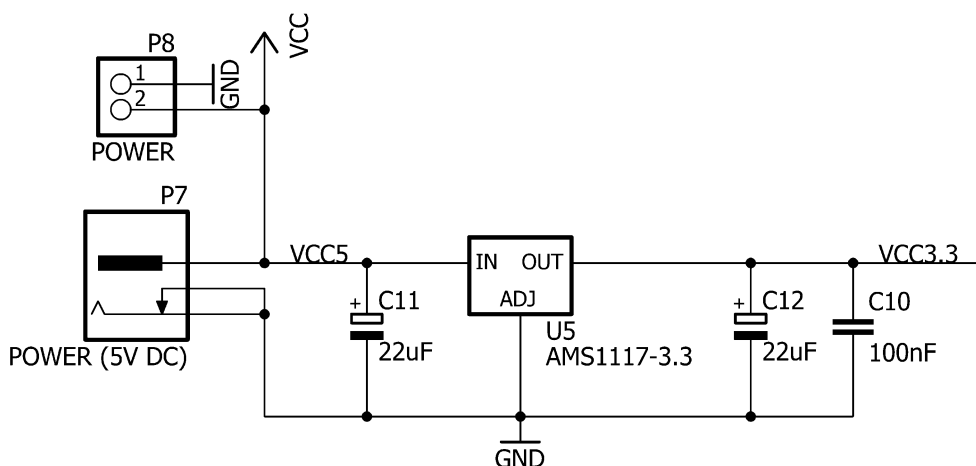
Jako zdroj hodinového signálu pro FPGA je použit oscilátor XO-7X5MM s frekvencí 50MHz.



Obrázek 2.15: Oscilátor XO-7X5MM-50MHz

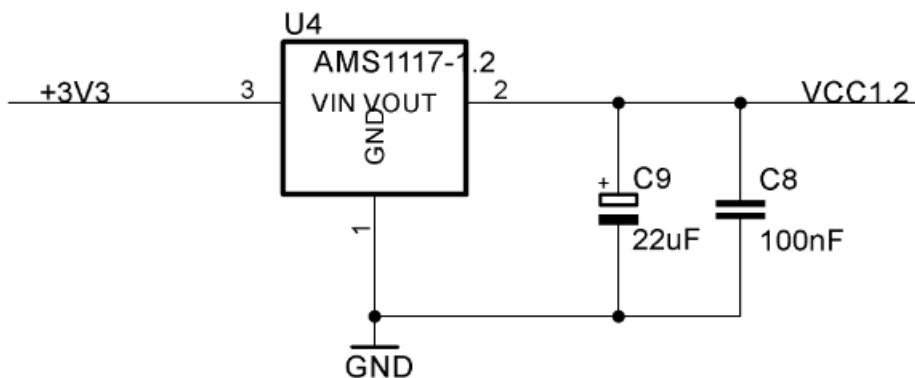
2.12 Napájení mikropočítačového řídicího systému

Napájení systému je rozděleno na digitální a analogové. Napájení digitálních částí systému je napájeno +5V zdrojem stejnosměrného napětí, které je připojeno na desku konektorem JACK. Napětí úroveň +5V slouží k napájení DC/DC měničů, bufferů a level shifterů. Dále je napětí stabilizováno pomocí stabilizátoru AMS1117-3.3 na napětí úroveň +3,3V. Tato úroveň je důležitá především pro napájení vstupů a výstupů programovatelného hradlového pole, viz obr. 2.16.



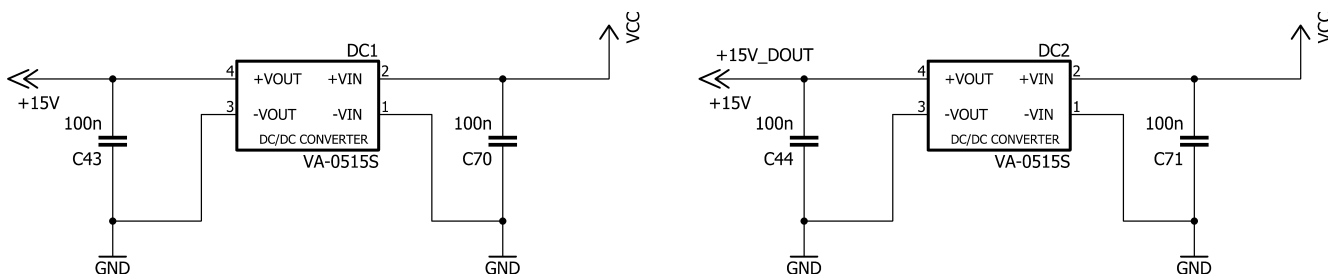
Obrázek 2.16: Schéma zapojení stabilizátoru AMS1117-3.3

Dále je zapotřebí stabilizátor ze +3,3V na +1,2V, viz obr. 2.17, protože napájení vstupně-výstupních buněk FPGA závisí na napětí úrovních I/O standardu, pomocí kterého komunikuje s okolím.



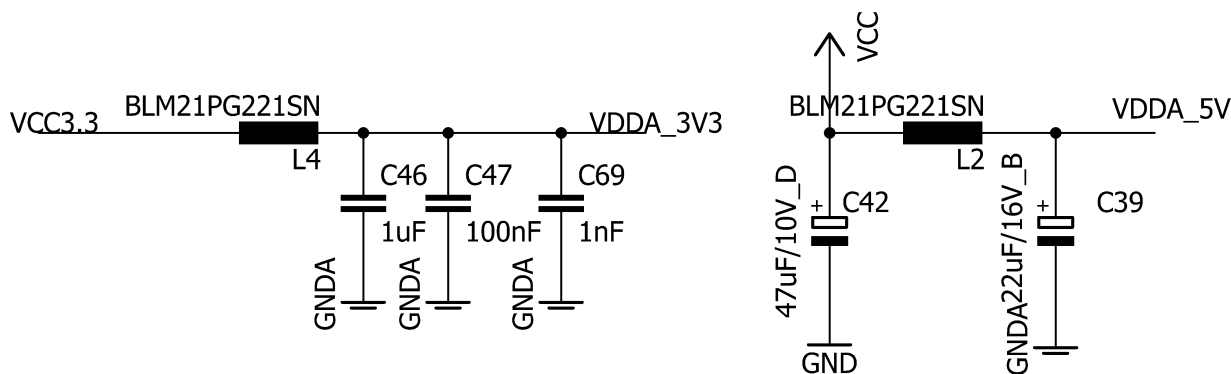
Obrázek 2.17: Schéma zapojení stabilizátoru AMS1117-1.2

Pro napájení obvodů, které potřebují k funkci napětíovou úroveň +15V, jsou na desce umístěny dva DC/DC měniče, které napětíovou úroveň +5V mění na +15V, viz obr. 2.18.



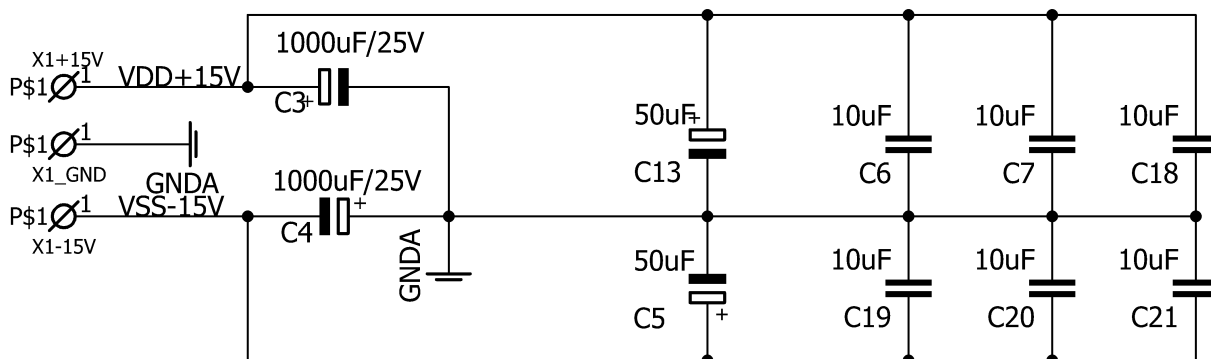
Obrázek 2.18: DC/DC měniče

Napájecí cesty pro obvody ADC jsou odrušeny feritovými perlami a filtrační kondenzátory.



Obrázek 2.19: Odrušení pro obvody ADC

Část analogového napájení je tvořena externím symetrickým zdrojem $\pm 15V$, který slouží pro napájení operačních zesilovačů. Napětí je přivedeno na desku konektory WAGO. Na desce je osazena filtrační kondenzátorová síť.



Obrázek 2.20: Přívod externího napájení $\pm 15V$

2.13 Programátor

Pro konfiguraci FPGA je systém vybaven programátorem USB Blaster od firmy ALTERA, viz obr. 2.21.



Obrázek 2.21: USB Blaster

2.14 Realizace mikropočítačového řídicího systému

Jak již bylo řečeno, mikropočítačový řídicí systém je složen ze dvou desek plošných spojů, kde vývojový kit eZdsp™ F28335 je propojen s rozšiřující deskou pomocí násuvných kolíků. Aby desky bylo možné používat samostatně, je každá deska vybavena vlastním přívodním konektorem JACK pro přivedení napájecího napětí. Je třeba brát na vědomí, že při spojení těchto dvou systémů je nutné připojit pouze jeden napájecí zdroj do jednoho z uvedených napájecích konektorů.

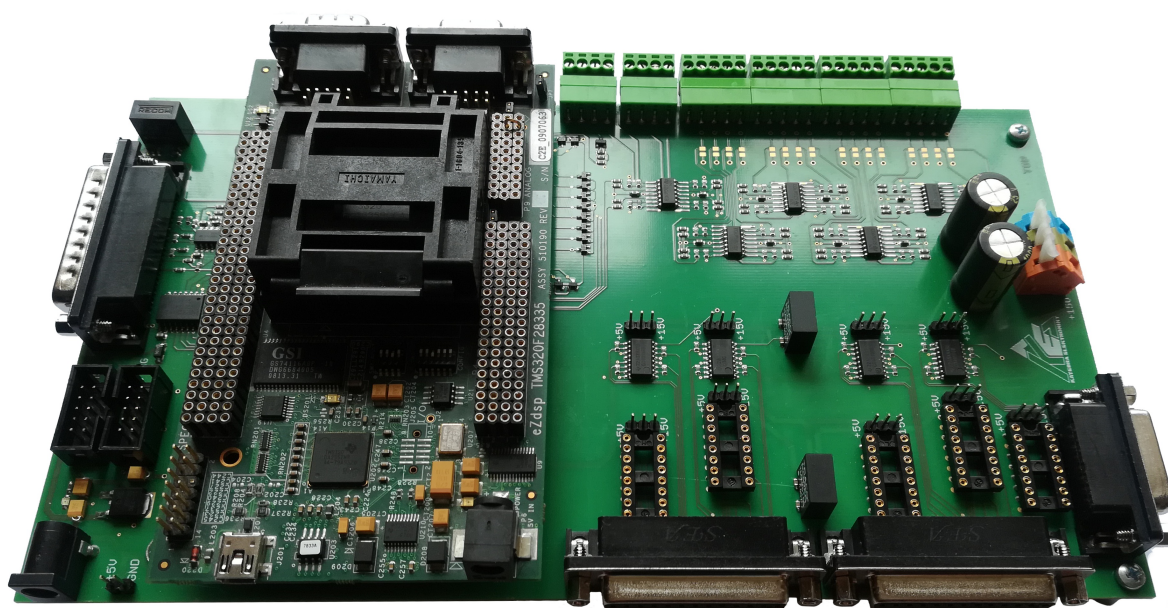
Rozšiřující deska s programovatelným hradlovým polem disponuje těmito parametry:

- Délka desky 24 cm
- Šířka desky 14 cm

Deska byla vyrobena čtyřvrstvou technologií, kde vnější vrstvy TOP a BOTTOM slouží jako signálové. Vnitřní vrstva SUPPLY slouží pro rozvod napájení a v poslední vrstvě GNDS jsou rozlité země. Deska byla vyrobena firmou Gatema v konstrukční třídě 6.

Tabulka 1.3: Konstrukční třída 6

Minimální šířka vodičů a mezer	0,15 mm
Minimální přesah plošky na poloměr	0,125 mm
Nejmenší průměr vrtání	0,20 mm



Obrázek 2.22: Realizace mikropočítačového řídicího systému

2.15 Popis I/O konektorů mikropočítačového řídicího systému

Pro snadnou orientaci uživatele mikropočítačového systému byl vytvořen tabulkový přehled zapojení I/O konektorů.

Tabulka 1.4: Analogové výstupy TMS

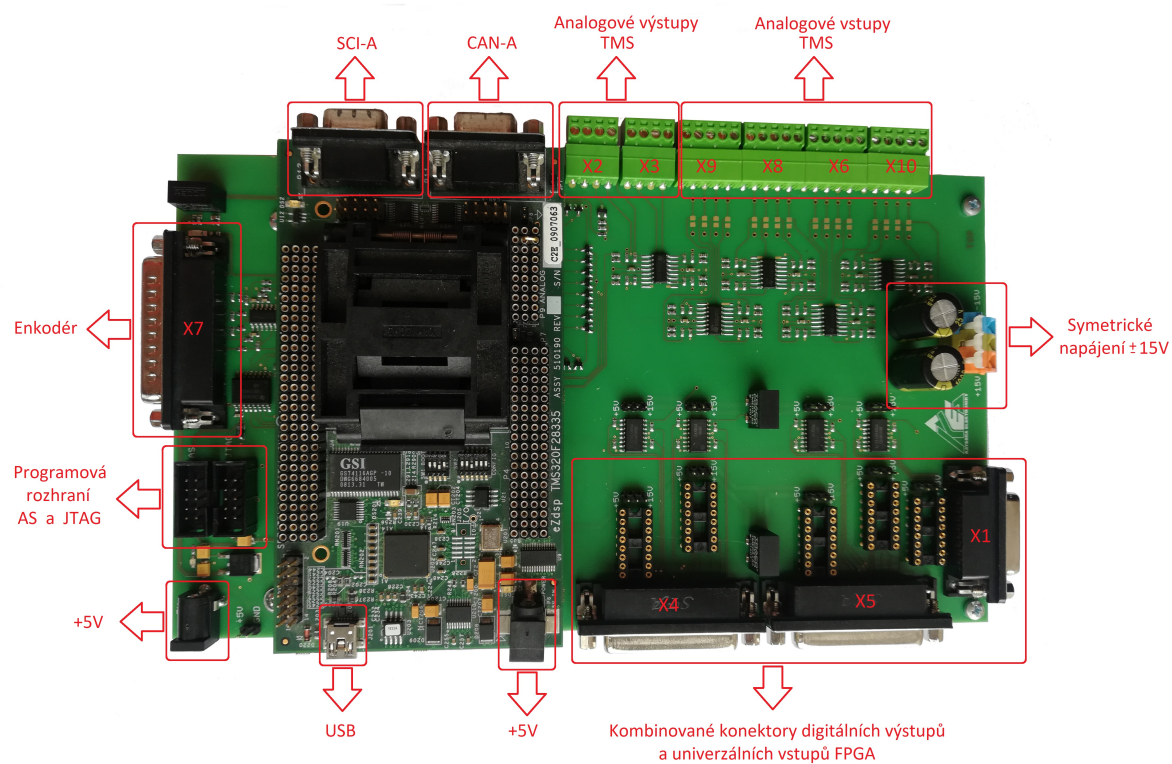
Pin	X2	X3
1	GND	GND
2	D/A_OUTB	D/A_OUTD
3	GND	GND
4	D/A_OUTA	D/A_OUTC

Tabulka 1.5: Analogové vstupy TMS

Pin	X6	X8	X9	X10
1	GND	GND	GND	GND
2	ADCINA0	ADCINA5	ADCINB3	ADCINA7
3	ADCINA1	ADCINB4	ADCINA3	ADCINB7
4	ADCINB1	ADCINB5	ADCINA2	ADCINA6
5	ADCINB0	ADCINA4	ADCINB2	ADCINB6

Tabulka 1.6: Kombinované konektory digitálních výstupů a univerzálních vstupů

Pin konektoru	Pin FPGA	X4	Pin FPGA	X5	Pin FPGA	X1
1	99	DOUT0	121	DOUT12	120	DOUT24
2	101	DOUT1	125	DOUT13	119	DOUT25
3	104	DOUT2	129	DOUT14	118	DOUT26
4	112	DOUT3	132	DOUT15	114	DOUT27
5	103	DOUT4	126	DOUT16	115	DOUT28
6	100	DOUT5	122	DOUT17	113	DOUT29
7	72	DOUT6	87	DOUT18		GND
8	75	DOUT7	93	DOUT19	-	-
9	79	DOUT8	96	DOUT20	9	INPUT_15
10	86	DOUT9	97	DOUT21	4	INPUT_12
11	76	DOUT10	94	DOUT22	7	INPUT_13
12	74	DOUT11	92	DOUT23	8	INPUT_14
13		GND		GND	-	-
14	-	-	-	-	-	GND
15	-	-	-	-	-	GND
16	136	INPUT_4	144	INPUT_10		
17	141	INPUT_3	143	INPUT_9		
18	134	INPUT_6	142	INPUT_8		
19	133	INPUT_5	3	INPUT_11		
20	GPIO21	INPUT_TMS	137	INPUT_1		
21	135	INPUT_7	139	INPUT_2		
22	-	-	-	-		
23	-	-	-	-		
24		GND		GND		
25		GND		GND		



Obrázek 2.23: Popis mikropočítačového řídicího systému

3 Základní programové vybavení

Základní programové vybavení bylo vytvořeno pro programovatelné hradlové pole i pro signálový procesor. FPGA i signálový procesor mají svá vlastní vývojová prostředí (IDE). Pro programovatelné hradlové pole je to Quartus II od firmy ALTERA, kde je konfigurace psána v jazyce VHDL. Pro signálový procesor se jedná o Code Composer Studio 8.0.0 od firmy Texas Instruments, kde je program psán v jazyce C. Všechny vytvořené programy budou k dispozici jako příloha na CD.

3.1 Programové vybavení signálového procesoru

Programové vybavení signálového procesoru obsahuje program pro komunikaci s programovatelným hradlovým polem. Program se skládá z cmd souboru *Zone6*, hlavičkového souboru *FPGA.h* a zdrojových souborů *FPGA.c* a *main.c*.

3.1.1 Cmd soubor pro paměťovou zónu 6

Pro komunikaci s externími zařízeními má signálový procesor TMS320F28335 tři paměťové zóny, a to zónu 0, zónu 6 a zónu 7. Pro zónu 0 je do budoucna plánováno připojení jiné periferie a zóna 7 je obsazena pro komunikaci s externí pamětí obsaženou přímo na kitu. Proto je pro komunikaci mikropočítačového systému využita zóna 6 s řídicím signálem CS6. Mapování paměťové zóny se nachází v cmd souboru *zone6*. V tomto souboru jsou popsány zóny v paměti a jejich sekce, kde každý registr má vlastní sekci, viz obr. 3.1.

```
MEMORY
{
  PAGE 0:      /* Program Memory */

  PAGE 1 :     /* Data Memory */
    ZONE6X000      : origin = 0x100000, length = 0x001      /* zone6 0X0000 */
    ZONE6X001      : origin = 0x100001, length = 0x001      /* zone6 0X0000 */
    ZONE6X002      : origin = 0x100002, length = 0x001      /* zone6 0X0000 */
}

SECTIONS
{
  zone6x000 : > ZONE6X000,      PAGE = 1
  zone6x001 : > ZONE6X001,      PAGE = 1
  zone6x002 : > ZONE6X002,      PAGE = 1
}
```

Obrázek 3.1: Cmd soubor *zone6*

3.1.2 Hlavičkový soubor

V souboru *FPGA.h* jsou deklarovány globální proměnné *reg0x*. Proměnné jsou deklarovány pomocí klíčového slova *extern*, což z *reg0x* dělá globální proměnnou, která je pak viditelná v celém programovém kódu. Hlavičkový soubor je zobrazen na obr. 3.2.

```

#ifndef INCLUDE_FPGA_H_
#define INCLUDE_FPGA_H_

// Globální proměnné
extern volatile Uint16 reg0x000;
extern volatile Uint16 reg0x001;
extern volatile Uint16 reg0x002;

#endif

```

Obrázek 3.2: Hlavičkový soubor *FPGA.h*

3.1.3 Zdrojové soubory

Zdrojový soubor *FPGA.c* slouží pro definici proměnných v jazyce C nebo jazyce C++. Určení definice pro typ jazyka je dáno podmínkou. Po zvolení definice programovacího jazyka je proměnná *reg0x* umístěna do zóny 6, viz obr. 3.3.

```

#include "DSP28x_Project.h"
#include "fpga.h"

// Globální proměnné

#ifdef __cplusplus
#pragma DATA_SECTION("zone6x000")
#else
#pragma DATA_SECTION(reg0x000, "zone6x000");
#endif
volatile Uint16 reg0x000;
#ifdef __cplusplus
#pragma DATA_SECTION("zone6x001")
#else
#pragma DATA_SECTION(reg0x001, "zone6x001");
#endif
volatile Uint16 reg0x001;
#ifdef __cplusplus
#pragma DATA_SECTION("zone6x002")
#else
#pragma DATA_SECTION(reg0x002, "zone6x002");
#endif
volatile Uint16 reg0x002;

```

Obrázek 3.3: Zdrojový soubor *FPGA.c*

Zdrojový soubor *main.c* slouží jako hlavní program. V první části programu se nachází vložení hlavičkových souborů, definice lokálních proměnných a deklarace lokálních funkcí potřebných k správnému běhu programu, viz obr. 3.4.


```

// Vložení hlavičkových souborů
#include "DSP28x_Project.h"
#include "fpga.h"

//Lokální proměnná
Uint16 u16LedDS2counter = 0;
Uint16 u16OutCount = 0;
Uint16 u16SigmaData0 = 0;
Uint16 u16SigmaData1 = 0;
float32 MP = 50.0; //procenta

//Deklarace lokálních funkcí
void initLedDS2(void);
void init_zone6(void);
interrupt void adc_isr(void);
interrupt void cpu_timer0_isr(void);

```

Obrázek 3.4: Zdrojový soubor main.c

Následují definice funkcí, které jsou podrobně popsány poznámkami přímo v kódu, viz obr. 3.5.

```

// Definice Funkcí
void main(void) {
    InitSysCtrl(); //Inicializace systemových součástí
    DINT; // Zakázání všech přerušení CPU
    InitPieCtrl(); // Inicializace PIE registru na výchozí stav.
    IER = 0x0000; // Zákaz přerušení CPU
    IFR = 0x0000; // Vynulování všech příznaků přerušení
    InitPieVectTable(); //Defaultní- tabulka vektorů přerušení
    // Přemapování vektorů přerušení se provede tímto zápisem:
    EALLOW;
    PieVectTable.TINT0 = &cpu_timer0_isr; Obsluha přerušení CPUtimeru
    PieVectTable.ADCINT = &adc_isr; //Obsluha přerušení od dokončení převodu
    EDIS;
    initLedDS2(); // Inicializace LED2
    init_zone6(); // Inicializace zóny 6
    InitCpuTimers(); // Inicializace CPUtimeru
    ConfigCpuTimer(&CpuTimer0, 150, 50);
    /* Povolení signálu přerušení do CPU*/
    IER |= M_INT1; // Povolení přerušení od skupiny 1
    // Povolení přerušení ve skupinách jednotky PIE
    PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Povolení přerušení 6. bitu ve
    PieCtrlRegs.PIEIER1.bit.INTx7 = 1; // skupině 1 (EOC ADC)
    // Povolení všech přerušení:
    EINT;
    // Povolení přerušení realtime beduggeru DBGEM
    ERTM;
    reg0x000 = 0; // Vynulování reg0
    reg0x001 = 0;
    StartCpuTimer0();
    // Nekonečná smyčka
    while (1) {

    }
}

```

Obrázek 3.5: Definice funkcí v main.c

Po definici funkcí program pokračuje obsluhou přerušení, kde je v přerušení časovače vytvořen blikáč pro vizuální kontrolu běhu programu. Dále se zde provádí zápis do registrů v reálném čase. Děje se tak pomocí funkce Debugger.

```
interrupt void cpu_timer0_isr(void) {
    CpuTimer0.InterruptCount++;
    u16LedDS2counter++;
    if (u16LedDS2counter > 5000) {
        u16LedDS2counter = 0;
        u16SigmaData0 = (Uint16)(MP * 0.01 * 0xFFFF);
        reg0x000 = u16SigmaData0;
        reg0x001 = u16SigmaData0;
        GpioDataRegs.GPBTOGGLE.bit.GPIO32 = 1;
    }
    // Acknowledge this interrupt to receive more interrupts from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
```

Obrázek 3.6: Přerušení `cpu_timer0` v `main.c`

Poslední částí programu je nastavení komunikace, pro kterou se využívá paměťová zóna 6. Nastavení je provedeno podle datasheetu *External Interface* [7]. Je nutno nastavit XINTF clock, který je jinak vypnutý z důvodu snížení spotřeby. Následuje nastavení GPIO registru pro XINTF, který musí být nastaven před použitím XINTF a dále je nastaveno časování pro zápis a čtení. Vše je nastaveno pro testovací účely na maximální hodnoty. Nastavení časování zápisu a čtení odpovídá časovým průběhům, viz obr. 3.8 a obr. 3.9.

```

void init_zone6(void) {
    // Make sure the XINTF clock is enabled
    EALLOW;
    SysCtrlRegs.PCLKCR3.bit.XINTFENCLK = 1; //
    EDIS;

    // Configure the GPIO for XINTF with a 16-bit data bus. This function is
    // in DSP2833x_Xintf.c
    InitXintf16Gpio();

    EALLOW;
    // All Zones-----
    XintfRegs.XINTCNF2.bit.XTIMCLK = 1; // Timing for all zones based on
    // TIMCLK = SYSCLKOUT/2
    XintfRegs.XINTCNF2.bit.WRBUFF = 3; // Buffer up to 3 writes
    XintfRegs.XINTCNF2.bit.CLKOFF = 1; // XCLKOUT is enabled
    XintfRegs.XINTCNF2.bit.CLKMODE = 0; // XCLKOUT = XTIMCLK
    // Disable XHOLD to prevent XINTF bus from going into high impedance
    // state whenever TZ3 signal goes low. This occurs because TZ3 on GPIO14
    // is shared with HOLD of XINTF
    XintfRegs.XINTCNF2.bit.HOLD = 1;
    // Zone 6-----
    // When using ready, ACTIVE must be 1 or greater, LEAD must always be 1
    // or greater
    // Zone write timing
    XintfRegs.XTIMING6.bit.XWRLEAD = 3;
    XintfRegs.XTIMING6.bit.XWRACTIVE = 7;
    XintfRegs.XTIMING6.bit.XWRTRAIL = 3;
    // Zone read timing
    XintfRegs.XTIMING6.bit.XRDLEAD = 3;
    XintfRegs.XTIMING6.bit.XRDACTIVE = 7;
    XintfRegs.XTIMING6.bit.XRDTRAIL = 3;
    XintfRegs.XTIMING6.bit.X2TIMING = 1; // don't double all Zone read/write
    // lead/active/trail timing

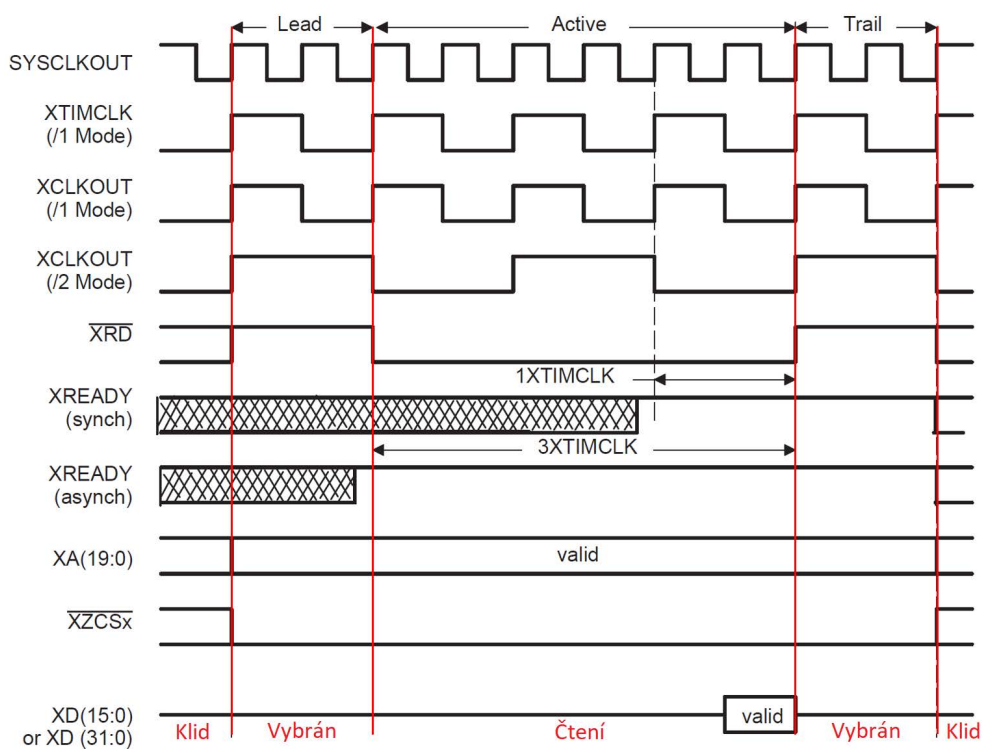
    // Zone will not sample XREADY signal
    XintfRegs.XTIMING6.bit.USEREADY = 0;
    XintfRegs.XTIMING6.bit.READYMODE = 0;
    XintfRegs.XTIMING6.bit.XSIZE = 3; // 1,1 = x16 data bus, 0,1 = x32 data
    // bus, other values are reserved

    EDIS;
    //Force a pipeline flush to ensure that the write to the last register
    // configured occurs before returning.
    __asm(" RPT #7 || NOP");
}

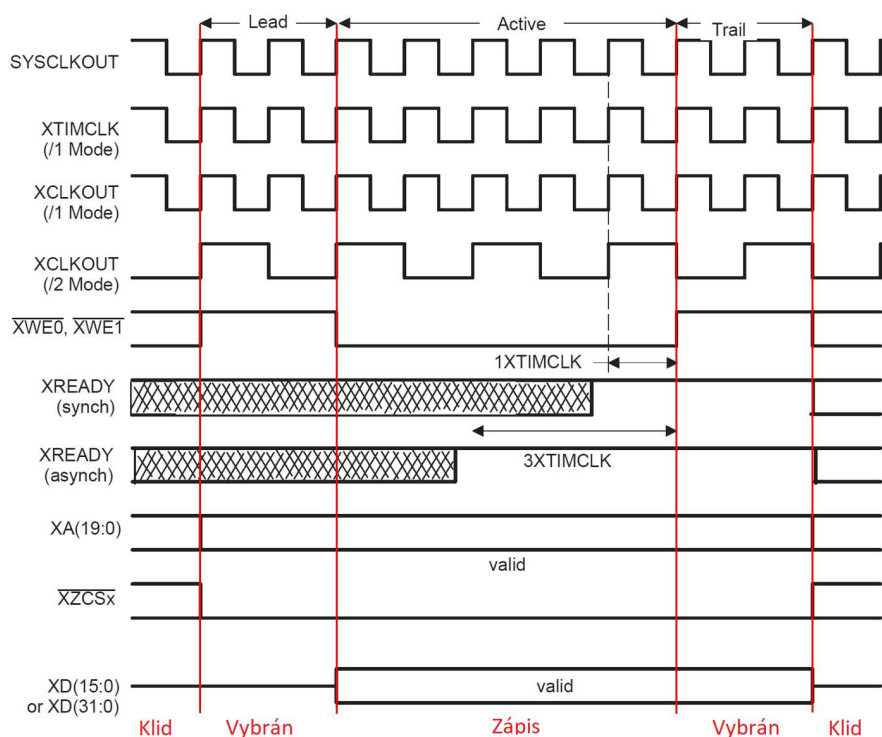
```

Obrázek 3.7: Nastavení zóny 6 v main.c

Podle časových průběhů pro čtení a zápis byl navržen stavový automat pro komunikaci s programovatelným hradlovým polem, který je následně popsán v kapitole 3.2.1.



Obrázek 3.8: Časové průběhy pro čtení



Obrázek 3.9: Časové průběhy pro zápis

3.2 Programové vybavení FPGA

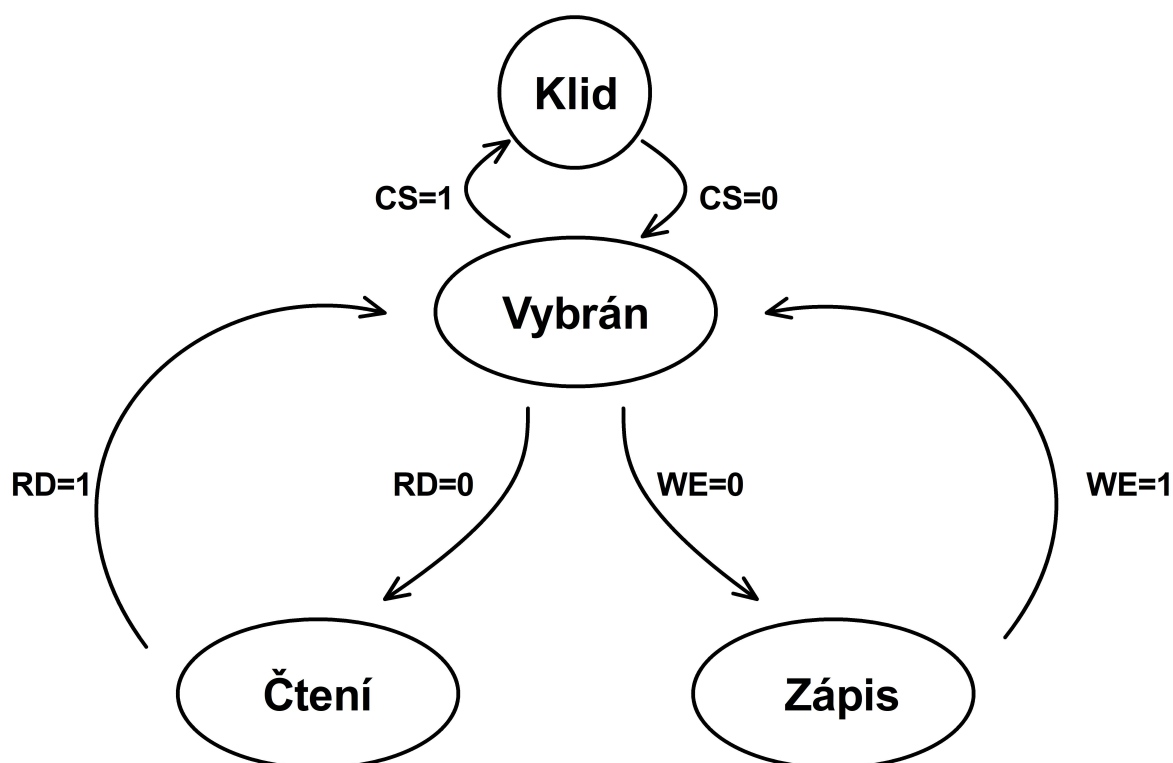
Programové vybavení pro FPGA je psáno v jazyce VHDL, který vychází z rodiny jazyků HDL (z angl. Hardware Description Language). Z názvu již vyplývá, že se jedná o jazyk popisný, nikoliv programovací. VHDL tedy popisuje vnitřní zapojení velmi rychlých integrovaných obvodů a digitálních zapojení. Lze tak popsat chování i složitých elektrických obvodů. Výhodou jazyka VHDL je, že disponuje prostředky pro popis paralelně probíhajících dějů, což v běžném programovacím jazyce, který probíhá sekvenčně, není možné. [10]

Pro FPGA byl vytvořen soubor konfiguračních programů, které zajišťují bezpečnou komunikaci řídicího systému. Dále byla vytvořena konfigurace sigma delta modulátoru pro ověření funkce řídicího systému.

3.2.1 Komunikace FPGA se signálovým procesorem

Jelikož FPGA a signálový procesor pracují s různým hodinovým signálem, je třeba synchronizovat hodinové domény. Synchronizace zajišťuje bezpečnou komunikaci po sběrnici, aby nedocházelo ke ztrátám informace a kolizním stavům. Synchronizace je prováděna synchronizačním taktem 300MHz.

Komunikační proces je nakonfigurován podle stavového automatu, viz obr. 3.10. Stavový automat je tvořen čtyřmi stavy: *Klid*, *Vybrán*, *Čtení*, *Zápis*. Stavy se mění na základě změn řídicích signálů *CS*, *WE* a *RD*.



Obrázek 3.10: Stavový automat pro komunikaci

Konfigurace je popsána behaviorálním procesem, ve kterém se operace vykonávají sekvenčně, podobně jako v jazyce C.

Proces je taktován na náběžnou hranu 300MHz hodinového signálu, kdy v každém taktu je prováděna kontrola stavů. Protože hradlového pole je napájeno hodinovým signálem CLK z oscilátoru 50MHz, je takt vynásoben pomocí interního fázového závěsu z knihovny Mega wizzard na 300MHz.

Aby byl zajištěn bezproblémový přechod mezi stavy, je do konfigurace zařazen signál *NovyStav*. Ten zajišťuje, aby nedocházelo k přepnutí probíhajícího stavu na jiný stav, čímž by mohlo dojít k zacyklení procesu.

Výchozí stav je *Klid*, z něhož se přechází do stavu *Vybrán* za předpokladu, že je splněna podmínka $CS=0$. Při přechodu do stavu *Vybrán* se nastaví *Data* do stavu *Z* (vysoká impedance). Dále se volí stav za následujících podmínek:

Pokud je signál $CS=1$ vrací se do stavu *Klid*. Pokud signál CS zůstává roven nule, testuje se, zda přichází signál pro aktivaci stavů *Čtení* nebo *Zápis*. Pokud je signál $RD=0$ přechází se do stavu *Čtení*. Tady se vykoná podprogram tvořený příkazem *case* pro určování adres, z kterých se čtou data. Pokud je signál $WE=0$ přechází se do stavu *Zápis*. Ve stavu *Zápis* se vykoná podprogram tvořený příkazem *case* pro určení adres, z kterých se data zapisují. Ze stavů *Čtení* nebo *Zápis* se přechází zpět do stavu *Vybrán* za podmínky, že řídicí signály $WE=1$ a $RD=1$. V tomto stavu se nastaví data do stavu *Z*. Celý proces je zobrazen na obr. 3.11.

```

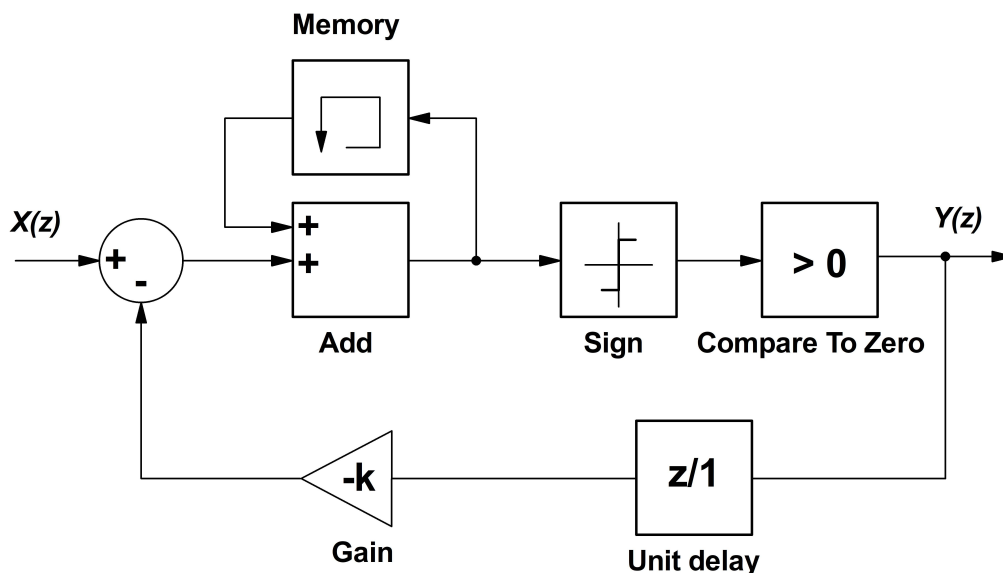
1  process (takt300MHz, Stav, CS, WE, RD) is
2  begin
3      if(rising_edge(takt300MHz)) then
4          Stav <= NovyStav;
5          case Stav is
6              when Klid =>
7                  if(CS = '0') then
8                      NovyStav <= Vybran;
9                  end if;
10                 Data <= "ZZZZZZZZZZZZZZZZ";
11             when Vybran =>
12                 if(CS = '1') then
13                     NovyStav <= Klid;
14                 else
15                     if(RD = '0') then
16                         NovyStav <= Cteni;
17                         case Adresy is
18                             when "0000000000" => Data <= std_logic_vector (D0);
19                             when "0000000001" => Data <= std_logic_vector (D1);
20                             when "0000000010" => Data <= std_logic_vector (D2);
21                             when others => null;
22                         end case;
23                     end if;
24                     if(WE = '0') then
25                         NovyStav <= Zapis;
26                         case Adresy is
27                             when "0000000000" => D0 <= unsigned(Data);
28                             when "0000000001" => D1 <= unsigned(Data);
29                             when others => null;
30                         end case;
31                     end if;
32                 end if;
33             when Cteni =>
34                 if(RD = '1') then
35                     NovyStav <= Vybran;
36                     Data <= "ZZZZZZZZZZZZZZZZ";
37                 end if;
38             when Zapis =>
39                 if(WE = '1') then
40                     NovyStav <= Vybran;
41                     Data <= "ZZZZZZZZZZZZZZZZ";
42                 end if;
43             when others => null;
44         end case;
45     end if;
46 end process;

```

Obrázek 3.11: Konfigurace komunikačního procesu

3.2.2 Sigma-delta modulátor 1. řádu

Programové vybavení obsahuje konfiguraci sigma-delta modulátoru, pomocí něhož je prováděno řízení snižujícího měniče napětí pro ověření funkčnosti mikropočítačového řídicího systému. Konfigurace sigma-delta modulátoru 1. řádu byla provedena dle navrhnutého matematického modelu, viz obr. 3.12.



Obrázek 3.12: Matematický model navrženého sigma-delta modulátoru 1. řádu

Konfigurace začíná výběrem standardních knihoven *ieee*, kde jsou definovány standardní pojmy potřebnými pro konfiguraci. Následuje popis *entity*, která uvozuje deklaraci. V entitě je pouze *port*, kde se deklaruje rozhraní sigma-delta modulátoru. V portu je deklarovaný jednobitový výstup *Q*, 16bitový vstup *D* a hodinový signál *CLK*. V části *architecture* je popsáno fungování obvodu (definice).

Za hlavičkou architektury jsou definovány lokální proměnné. Zde je definován 18bitový signál *suma*, jednobitový signál *vystup* a konstanta *PulRozsahu*. Výkonná část programu začíná za slovem *begin*. Tato část je popsána procesem, který reaguje na signály *CLK*, *D* a *vystup*. Proces se spustí nástupnou hranou hodinového signálu *CLK*. Následuje podmínka, že pokud je signál *vystup* = 1, tak se do signálu *suma* zapíše součet stávající sumy se signálem *D*, od něhož je odečtena konstanta *PulRozsahu*. Pro bezproblémové provedení matematických operací je provedena úprava datové šířky 16bitového signálu *D* na 18bitový, protože signál *suma* a konstanta *PulRozsahu* jsou 18bitové. Následuje podmínka pro porovnávání signálu *suma* s nulou. Pokud je signál *suma* větší než nula, je signál *vystup* = 1. Pokud není signál *suma* větší než nula, je signál *vystup* = 0. V poslední části procesu se signál *vystup* zapíše na výstup modulátoru *Q*.


```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity SDM is
6      port (
7          Q: out std_logic;
8          D: in unsigned(15 downto 0);
9          Clk: in std_logic
10     );
11 end entity;
12
13 architecture first of SDM is
14     signal suma: signed (17 downto 0) := (others=>'0');
15     signal vystup: std_logic := '0';
16     constant PulRozsahu: signed(17 downto 0) := "010000000000000000";
17 begin
18     process (Clk, D, vystup) is
19     begin
20         if (rising_edge(Clk)) then
21             if(vystup = '1') then
22                 suma <= suma + (signed("00" & D) - PulRozsahu);
23             else
24                 suma <= suma + signed("00" & D);
25             end if;
26             if(suma > 0) then
27                 vystup <= '1';
28             else
29                 vystup <= '0';
30             end if;
31         end if;
32         Q <= vystup;
33     end process;
34 end architecture;

```

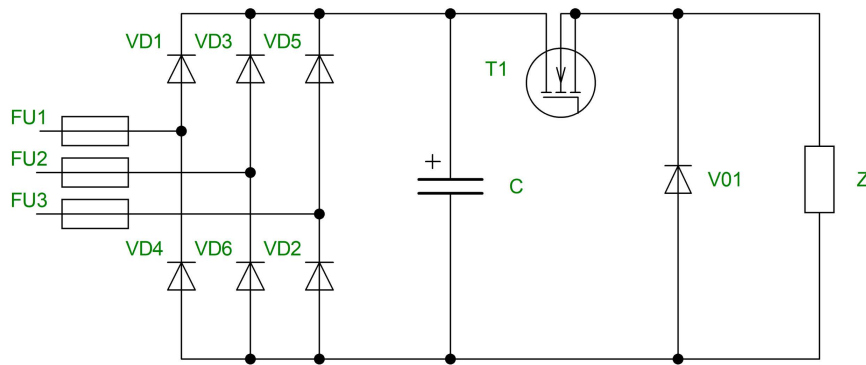
Obrázek 3.13: Konfigurace sigma-delta modulátoru

3.2.3 Pin planner

Nutnou součástí programového vybavení je pin planner, který slouží k propojení programových vstupů a výstupů s těmi fyzickými. Nastavení pin planneru bude umístěno v příloze.

4 Ověření funkce řídicího systému se sigma delta modulátorem

Pro ověření funkce byl řídicí systém připojen ke snižujícímu měniči napětí s RL zátěží, viz obr. 4.1. Výstupní napětí měniče bylo řízeno sigma-delta modulací z navrženého sigma-delta modulátoru 1. řádu.



Obrázek 4.1: Schéma snižujícího měniče napětí

Na snižujícím měniči byla proměřena závislost výstupního napětí měniče na modulačním poměru navrženého sigma-delta modulátoru. Dále byl měřen výstupní proud měniče a frekvence. Naměřené hodnoty jsou zpracovány v tabulce 4.1 spolu s vypočtenými hodnotami.

Tabulka 4.1: Naměřené hodnoty závislosti na modulačním poměru

	MP [%]	0	5	10	15	20	25	30	35	40	45	50
Naměřeno	U_d [V]	110	110	109	108	107,5	106,7	105,8	104,9	104	103,2	102,5
	U_{out} [V]	0	4,9	10,4	15,8	21,15	26,4	31,8	36,4	41,8	45,9	51,2
	I_{out} [A]	0	0,19	0,4	0,6	0,85	1	1,25	1,46	1,63	1,84	2
	f [KHz]	0	2,5	5	7,7	10	12,5	14,3	16,7	20	14,3	16,7
Teoreticky vypočteno	MP_{vyp} [%]	0	4,45	9,54	14,63	19,67	24,74	30,06	34,7	40,19	44,48	49,95
	$U_{out(vyp)}$ [V]	0	5,5	10,9	16,2	21,5	26,7	31,7	36,7	41,6	46,4	51,2
	X [%]	-	10,91	4,59	2,47	1,63	1,03	-0,19	0,86	-0,48	1,16	0,1
	MP [%]	55	60	65	70	75	80	85	90	95	100	
Naměřeno	U_d [V]	101,9	101,1	100,6	99,9	99,3	98,5	97,9	97,2	96,4	95,8	
	U_{out} [V]	55,6	60,7	64,9	69,5	74	78,2	82,3	86,5	90,5	94	
	I_{out} [A]	2,2	2,4	2,6	2,8	3	3,15	3,3	3,5	3,65	3,8	
	f [KHz]	14,3	19	16,7	14,3	12,5	9,8	7,6	5	2,5	0	
Teoreticky vypočteno	MP_{vyp} [%]	54,56	60,04	64,51	69,57	74,52	79,39	84,07	88,99	93,88	98,12	
	$U_{out(vyp)}$ [V]	56	60,7	65,4	69,9	74,5	78,8	83,2	87,5	91,6	95,8	
	X [%]	0,79	-0,07	0,75	0,61	0,64	0,76	1,10	1,12	1,18	1,88	

Vypočtené hodnoty modulačního poměru MP jsou počítány z naměřených hodnot stejnosměrného napětí v meziobvodu U_d a výstupního napětí měniče U_{out} . Dále byla vypočtena modulační chyba, která určuje podíl modulační odchylky (rozdíl mezi žádaným modulačním poměrem a skutečným/vypočteným) a skutečného modulačního poměru. Vypočtené výstupní napětí $U_{out(vyp)}$ pak vychází z teoretického vztahu pro velikost modulačního poměru.

Z tabulky 4.1 lze pozorovat, že napětí ve stejnosměrném meziobvodu U_d klesá se zatížením měniče, což je způsobeno poklesem napětí napájecího zdroje. Pokles je kompenzován ve výpočtu.

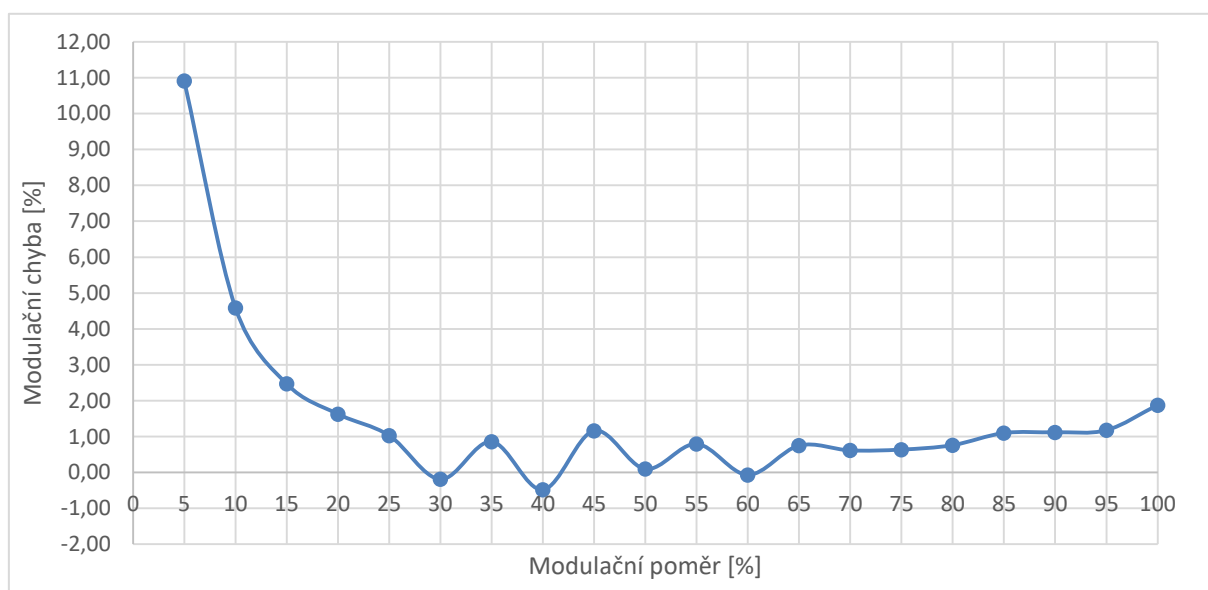
Příklady výpočtů:

$$MP = \frac{U_{out}}{U_d} = \frac{51,2}{102,5} \cdot 100 = 49,95 \% \quad (4.1)$$

$$X = 100 - \left(\frac{MP_{vyp}}{MP} \cdot 100 \right) = 100 - \left(\frac{49,95}{50} \cdot 100 \right) = 0,1 \% \quad (4.2)$$

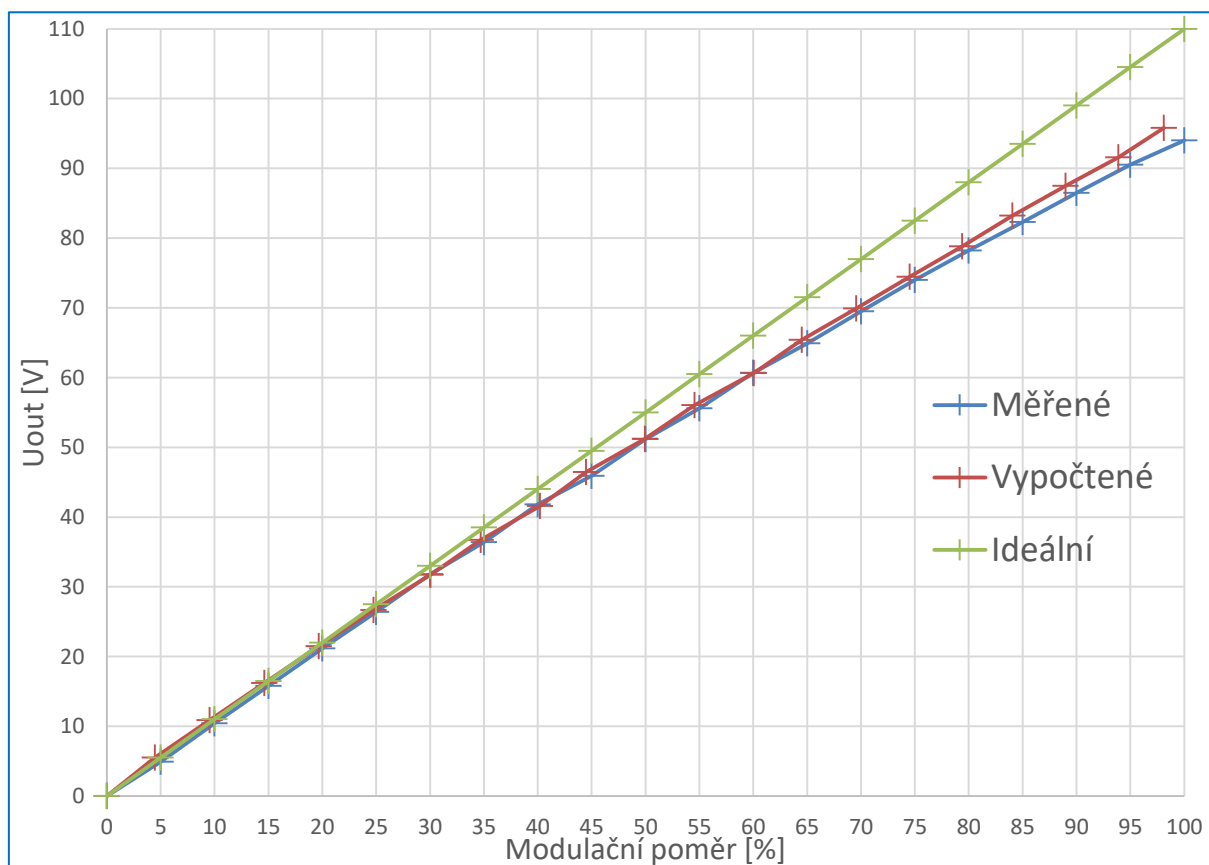
$$U_{out(vyp)} = U_d \cdot \frac{MP}{100} = 110 \cdot \frac{50}{100} = 49,95 V \quad (4.3)$$

Modulační chyba je vynesena do grafu, který je zobrazen na obr. 4.2. Největší modulační chybu lze pozorovat při 5% modulačního poměru. Chyba je způsobena chybou měření ve zpětné vazbě, kdy při malých hodnotách není měření dostatečně přesné. S rostoucím modulačním poměrem, tedy i s většími hodnotami výstupního napětí, lze pozorovat, že se modulační chyba pohybuje kolem 1%.



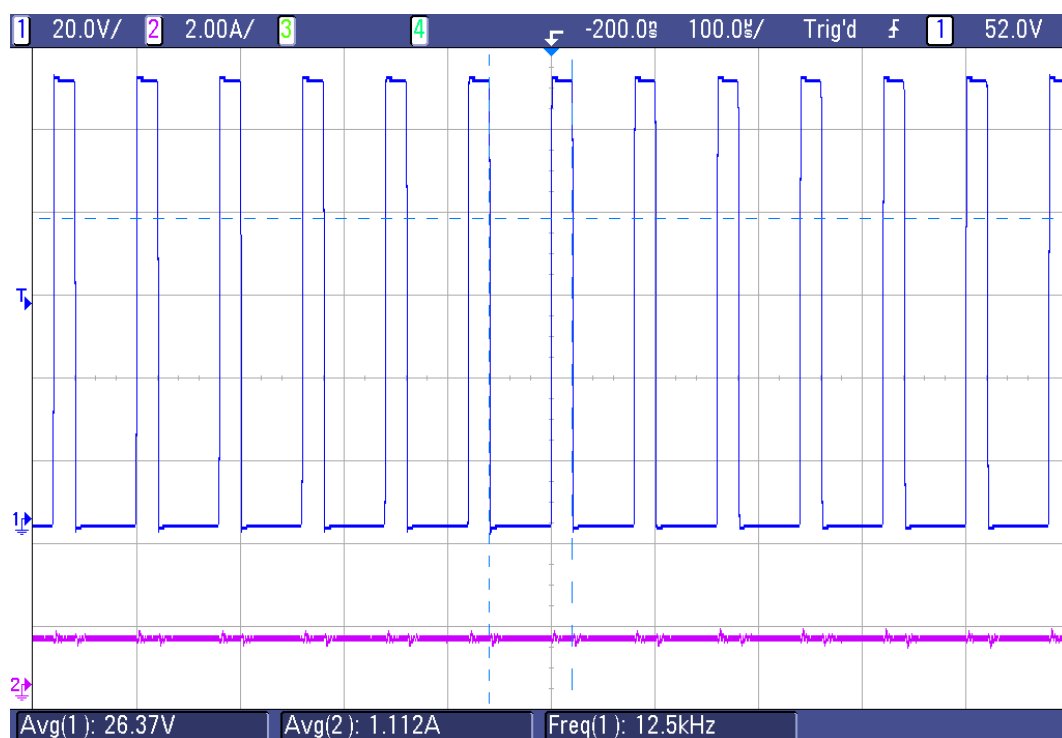
Obrázek 4.2: Modulační chyba

Závislost výstupního napětí na modulačním poměru je znázorněna grafem, viz obr. 4.3. V grafu jsou vyneseny tři závislosti výstupního napětí na modulačním poměru, a to ideální, měřená a vypočtená. Z grafu lze pozorovat pokles měřeného výstupního napětí, což je způsobeno tím, že při vyšším zatížení klesá vstupní napětí v meziobvodu měniče. Vypočtená hodnota výstupního napětí uvažuje i vliv poklesu v meziobvodu. Rozdíl mezi ní a naměřenou hodnotou je minimální a lze jej přisoudit nepřesnosti měření.

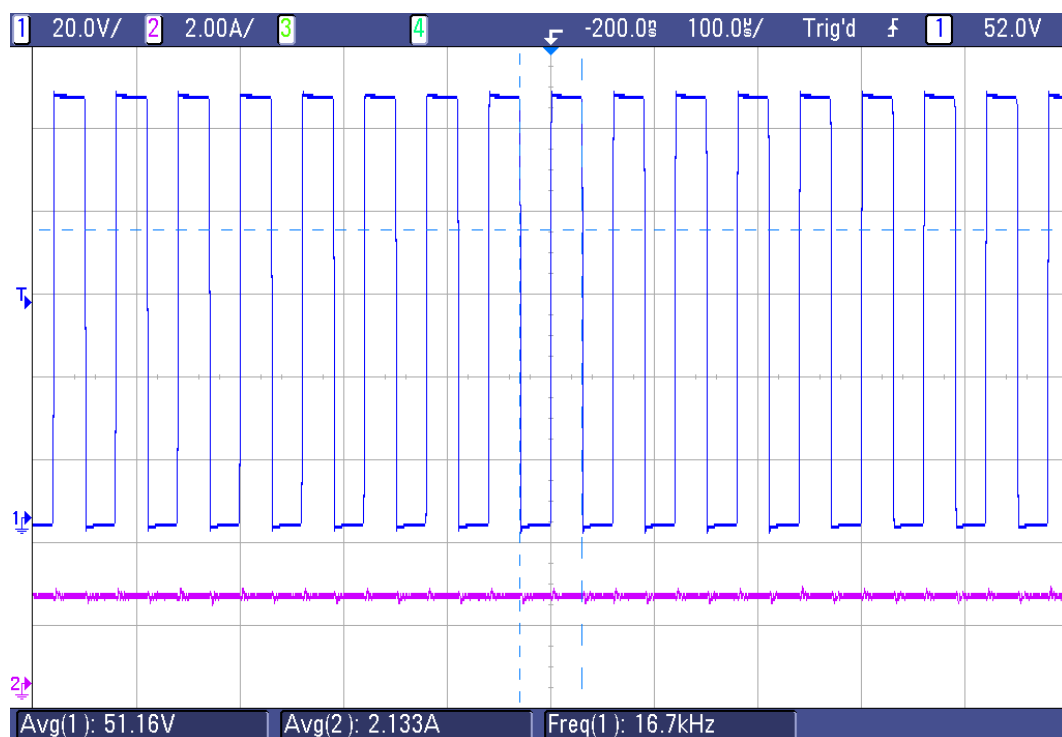


Obrázek 4.3: Závislost výstupního napětí snižujícího měniče na modulačním poměru (modrý průběh měřený, červený průběh vypočtený, zelený průběh ideální)

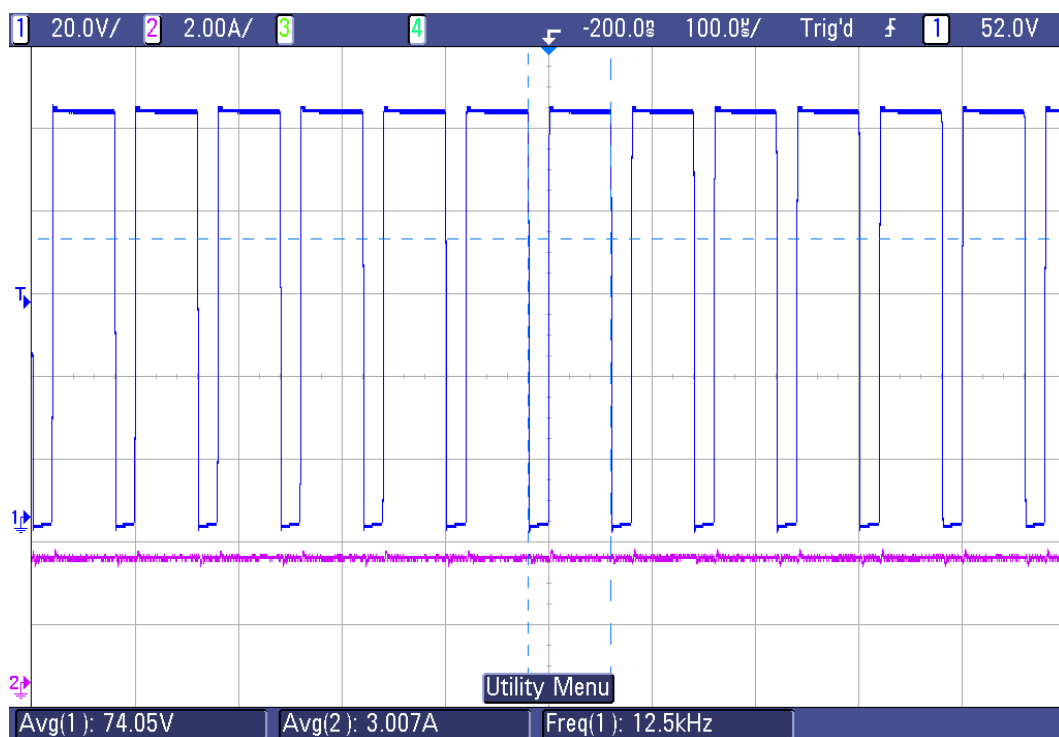
Naměřené průběhy výstupního napětí a proudu jsou zobrazeny na obr. 4.4 - 4.7. Na obr. 4.7 je zobrazen průběh výstupního napětí a proudu pro modulační poměr $MP = 15\%$, kde lze pozorovat, jak se mění spínací frekvence, což potvrzuje funkci sigma-delta modulace. Překrytí pulzu je způsobeno tím, že se nemění šířka pulzu, ale mění se perioda signálu.



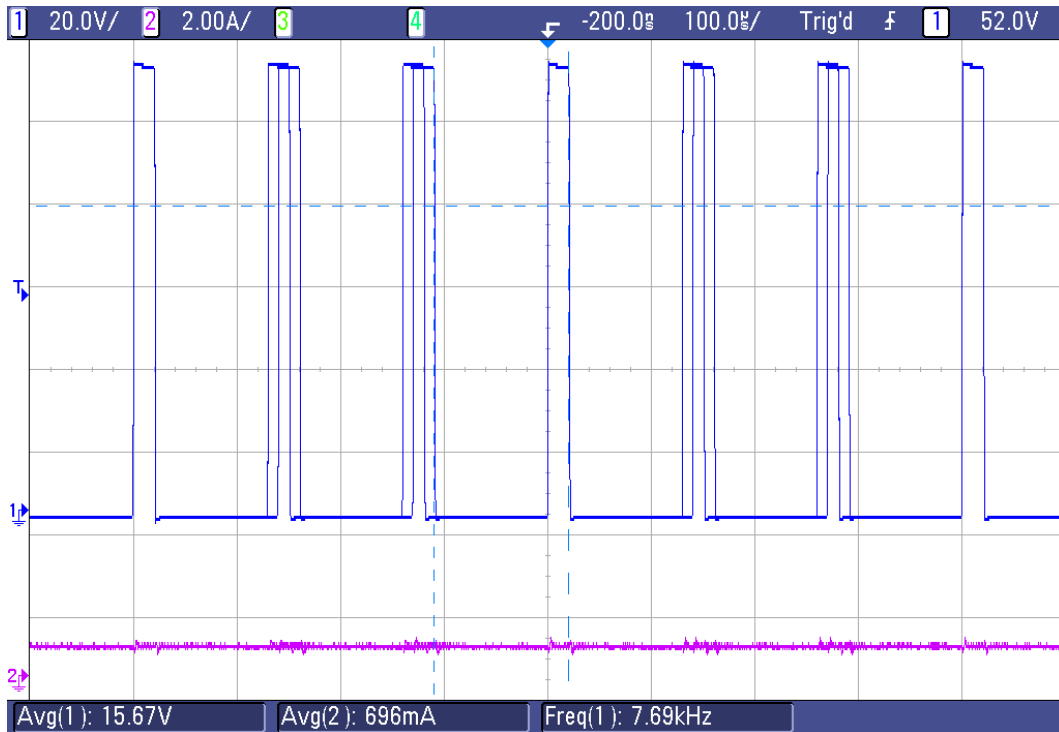
Obrázek 4.4: Průběh výstupního napětí (modrý průběh) a proudu (fialový průběh) pro MP=25%



Obrázek 4.5: Průběh výstupního napětí (modrý průběh) a proudu (fialový průběh) pro MP=50%



Obrázek 4.6: Průběh výstupního napětí (modrý průběh) a proudu (fialový průběh) pro MP=75%



Obrázek 4.7: Průběh výstupního napětí (modrý průběh) a proudu (fialový průběh) pro MP=15%

Závěr

Cílem této diplomové práce bylo navrhnout a následně realizovat mikropočítačový řídicí systém se sigma-delta modulátorem pro řízení výkonových měničů.

První kapitola se zabývá teoretickým rozбором topologií sigma-delta modulátorů vhodných pro řízení výkonových měničů. Byly zde také vysvětleny základní pojmy potřebné pro pochopení sigma-delta modulátoru obecně. Druhá kapitola je zaměřena na popis jednotlivých bloků mikropočítačového řídicího systému a zdůvodnění jejich použití. Dále se věnuje popisu realizovaného mikropočítačového systému. Ve třetí kapitole je popsáno programové vybavení potřebné pro komunikaci signálového procesoru s programovatelným hradlovým polem. Dále je v této kapitole detailně popsána konfigurace sigma-delta modulátoru, který je konfigurován jazykem VHDL. Poslední kapitola se věnuje ověření funkce mikropočítačového řídicího systému se sigma-delta modulátorem. Ověření bylo provedeno připojením mikropočítačového řídicího systému ke snižujícímu měniči napětí, kde bylo provedeno měření.

Tato práce přinesla funkční mikropočítačový řídicí systém, který je díky programovatelnému hradlovému poli schopen pracovat se signály o vysokých frekvencích. Toho lze využít pro řízení rezonančních měničů, tedy měničů s měkkým spínáním. Spínání těchto měničů je tedy velmi rychlé a samotný procesor by nebyl schopen tak rychlé výpočty zpracovat. Dále řídicí systém disponuje velkým počtem digitálních výstupů, což je vhodné pro řízení složitějších struktur, které mají velký počet spínacích tranzistorů, jako jsou např. víceúrovňové měniče.

Do budoucna lze systém vylepšit použitím modernějšího a výkonnějšího signálového procesoru a programovatelného hradlového pole.

Použitá literatura








- [1] VRBA, Kamil a David KUBÁNEK, 2014. *A/D a D/A převodníky pro integrovanou výuku VUT a VŠB-TUO* [online]. Brno [cit. 2018-04-27]. ISBN 978-80-214-5116-2.
- [2] ŠIŠKA, Martin. *Impulzové modulace*. Brno, 2013. Diplomová práce na fakultě elektrotechniky a komunikačních technologií Vysokého učení technického. Ústav telekomunikací. Vedoucí diplomové práce Radim Číť.
- [3] SCHREIER, Richard a Gabor C. TEMES. *Understanding Delta-Sigma Data Converters*. Hoboken: Wiley, 2005. 446 s. ISBN 0-471-46585-2.
- [4] MACHÁČEK, Zdeněk a Pavel NEVŘIVA. *Modulované signály: Učební text*. 1 vyd. Ostrava: VŠB-TU, 2012. 267 s. ISBN 978-80-248-2600-4.
- [5] BAKER, Bonnie. How the SNRs of delta-sigma converters differ. *EDN*. [online]. Aug 02 2007 [cit. 27.4.2018]. Dostupné z: <http://www.edn.com/electronics-blogs/bakers-best/4317368/How-the-SNRs-of-delta-sigma-converters-differ>
- [6] ANDREANI, Pietro, Advanced AD/DA converters. *Www.eit.lth.se* [online]. [cit. 2018-04-27]. Dostupné z: <https://www.eit.lth.se/fileadmin/eit/courses/etin05/2012/lec3.pdf>
- [7] *Datasheet: TMS320x2833x, 2823x DSC External Interface XINTF* [online], [cit. 2018-04-27]. Dostupné z: <http://www.ti.com/lit/ug/spru949d/spru949d.pdf>
- [8] *Datasheet: EZdspTM F28335 Technical Reference: Appendix A eZdspTM F28335 Schematics*. Dostupné z: <http://www.ti.com/lit/ug/spruhs1a/spruhs1a.pdf>
- [9] VOJÁČEK, Antonín, *Rotační enkodéry pro řízení pohonů a pohybu* [online]. 2017 [cit. 2018-04-27]. Dostupné z: <https://automatizace.hw.cz/rotacni-enkodery-pro-řízení-pohonu-a-pohybu.html>
- [10] *Učíme se VHDL* [online]. [cit. 2016-04-12]. Dostupné z: <http://vhdl.cz/uplne-zaklady-a-nezbytna-teorie>

Seznam příloh

Příloha A: Pin Planner

Příloha na CD/DVD.

Adresářová struktura přiloženého CD/DVD

-  Datasheet_ezdspf28335
-  DP_Dvo0143
-  DPS_EAGLE
-  Foto
-  Měření
-  Programové vybavení
-  Seznam součástek

Příloha A: *Pin planner*

To	Direction	Location	I/O Bank	VREF Group	Fitter Location
Adresy[9]	Input	PIN_31	1	B1_N1	PIN_31
Adresy[8]	Input	PIN_30	1	B1_N1	PIN_30
Adresy[7]	Input	PIN_40	4	B4_N1	PIN_40
Adresy[6]	Input	PIN_32	1	B1_N1	PIN_32
Adresy[5]	Input	PIN_42	4	B4_N1	PIN_42
Adresy[4]	Input	PIN_41	4	B4_N1	PIN_41
Adresy[3]	Input	PIN_44	4	B4_N1	PIN_44
Adresy[2]	Input	PIN_43	4	B4_N1	PIN_43
Adresy[1]	Input	PIN_47	4	B4_N1	PIN_47
Adresy[0]	Input	PIN_45	4	B4_N1	PIN_45
Data[15]	Bidir	PIN_51	4	B4_N1	PIN_51
Data[14]	Bidir	PIN_48	4	B4_N1	PIN_48
Data[13]	Bidir	PIN_52	4	B4_N1	PIN_52
Data[12]	Bidir	PIN_53	4	B4_N1	PIN_53
Data[11]	Bidir	PIN_55	4	B4_N1	PIN_55
Data[10]	Bidir	PIN_57	4	B4_N1	PIN_57
Data[9]	Bidir	PIN_58	4	B4_N1	PIN_58
Data[8]	Bidir	PIN_59	4	B4_N0	PIN_59
Data[7]	Bidir	PIN_60	4	B4_N0	PIN_60
Data[6]	Bidir	PIN_63	4	B4_N0	PIN_63
Data[5]	Bidir	PIN_64	4	B4_N0	PIN_64
Data[4]	Bidir	PIN_65	4	B4_N0	PIN_65
Data[3]	Bidir	PIN_67	4	B4_N0	PIN_67
Data[2]	Bidir	PIN_69	4	B4_N0	PIN_69
Data[1]	Bidir	PIN_70	4	B4_N0	PIN_70
Data[0]	Bidir	PIN_71	4	B4_N0	PIN_71
Din[14]	Input	PIN_9	1	B1_N0	PIN_9
Din[13]	Input	PIN_8	1	B1_N0	PIN_8
Din[12]	Input	PIN_7	1	B1_N0	PIN_7
Din[11]	Input	PIN_4	1	B1_N0	PIN_4
Din[10]	Input	PIN_3	1	B1_N0	PIN_3
Din[9]	Input	PIN_144	2	B2_N1	PIN_144
Din[8]	Input	PIN_143	2	B2_N1	PIN_143
Din[7]	Input	PIN_142	2	B2_N1	PIN_142
Din[6]	Input	PIN_135	2	B2_N1	PIN_135
Din[5]	Input	PIN_134	2	B2_N1	PIN_134
Din[4]	Input	PIN_133	2	B2_N1	PIN_133

Din[3]	Input	PIN_136	2	B2_N1	PIN_136
Din[2]	Input	PIN_141	2	B2_N1	PIN_141
Din[1]	Input	PIN_139	2	B2_N1	PIN_139
Din[0]	Input	PIN_137	2	B2_N1	PIN_137
Dout[29]	Output	PIN_113	2	B2_N0	PIN_113
Dout[28]	Output	PIN_115	2	B2_N0	PIN_115
Dout[27]	Output	PIN_114	2	B2_N0	PIN_114
Dout[26]	Output	PIN_118	2	B2_N0	PIN_118
Dout[25]	Output	PIN_119	2	B2_N0	PIN_119
Dout[24]	Output	PIN_120	2	B2_N0	PIN_120
Dout[23]	Output	PIN_92	3	B3_N0	PIN_92
Dout[22]	Output	PIN_94	3	B3_N0	PIN_94
Dout[21]	Output	PIN_97	3	B3_N0	PIN_97
Dout[20]	Output	PIN_96	3	B3_N0	PIN_96
Dout[19]	Output	PIN_93	3	B3_N0	PIN_93
Dout[18]	Output	PIN_87	3	B3_N1	PIN_87
Dout[17]	Output	PIN_122	2	B2_N0	PIN_122
Dout[16]	Output	PIN_126	2	B2_N0	PIN_126
Dout[15]	Output	PIN_132	2	B2_N1	PIN_132
Dout[14]	Output	PIN_129	2	B2_N1	PIN_129
Dout[13]	Output	PIN_125	2	B2_N0	PIN_125
Dout[12]	Output	PIN_121	2	B2_N0	PIN_121
Dout[11]	Output	PIN_74	3	B3_N1	PIN_74
Dout[10]	Output	PIN_76	3	B3_N1	PIN_76
Dout[9]	Output	PIN_86	3	B3_N1	PIN_86
Dout[8]	Output	PIN_79	3	B3_N1	PIN_79
Dout[7]	Output	PIN_75	3	B3_N1	PIN_75
Dout[6]	Output	PIN_72	4	B4_N0	PIN_72
Dout[5]	Output	PIN_100	3	B3_N0	PIN_100
Dout[4]	Output	PIN_103	3	B3_N0	PIN_103
Dout[3]	Output	PIN_112	2	B2_N0	PIN_112
Dout[2]	Output	PIN_104	3	B3_N0	PIN_104
Dout[1]	Output	PIN_101	3	B3_N0	PIN_101
Dout[0]	Output	PIN_99	3	B3_N0	PIN_99
RD	Input	PIN_25	1	B1_N1	PIN_25
WE	Input	PIN_24	1	B1_N1	PIN_24
CS	Input	PIN_28	1	B1_N1	PIN_28
Clk	Input	PIN_17	1	B1_N0	PIN_17